

**UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR**



ESTIMACIÓN DE FIABILIDAD DEL SEGUIMIENTO DE OBJETOS EN VÍDEO

-PROYECTO FIN DE CARRERA-

**Álvaro Calvo Tapia
Julio 2014**

ESTIMACIÓN DE FIABILIDAD DEL SEGUIMIENTO DE OBJETOS EN VÍDEO

Autor: Álvaro Calvo Tapia

Tutor: Juan Carlos San Miguel Avedillo

Ponente: José María Martínez Sánchez

email: {alvaro.calvo@estudiante.uam.es, Juancarlos.Sanmiguel@uam.es, JoseM.Martinez@uam.es}



Video Processing and Understanding Lab

Departamento de Tecnología Electrónica y de las Comunicaciones

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Julio 2014

Trabajo parcialmente financiado por el gobierno español bajo el proyecto TEC2011-25995 (Event Video)



Resumen

En este proyecto se propone una técnica para estimar la fiabilidad de los algoritmos de seguimiento de objetos en vídeo (*trackers*). Esta estimación consiste en determinar durante la ejecución del *tracker* (*online*) y en ausencia de los datos de *ground-truth* (anotaciones manuales de los resultados de seguimiento ideales) los instantes en los que el algoritmo sigue al objeto deseado de manera satisfactoria.

En primer lugar se lleva a cabo un estudio del estado del arte referente a la estimación de fiabilidad en el que se analizan algunas de las técnicas existentes. Después el proyecto se centra en la descripción del algoritmo propuesto cuyo principal objetivo es detectar los *frames* de cambio en los que el *tracker* pierde o recupera al objeto. Para ello se hace uso de un conjunto de características relacionadas con la forma, el movimiento y la apariencia del objeto seguido. En los instantes en los que estas características experimentan variaciones repentinas se considera la existencia de un *frame* de cambio. Para identificar estos valores atípicos de las características se plantea una estrategia de detección de anomalías. Posteriormente se emplea una máquina de estados para decidir en cada *frame* si el seguimiento es correcto o incorrecto (fiabilidad estimada).

Por último se evalúa el algoritmo propuesto sobre seis *trackers* distintos y se compara con las principales técnicas relacionadas. Para ello se ha seleccionado un *dataset* que incluye los problemas más comunes en el seguimiento de objetos.

Palabras clave

Seguimiento de objetos en vídeo, estimación de fiabilidad, evaluación *online*, *frames* de cambio, características, detección de anomalías.

Abstract

In this master thesis we propose an approach to estimate the reliability of video tracking algorithms. This estimation consists on determining during online analysis and in the absence of ground truth data (manual annotations of the ideal tracking results) the instants in which the algorithm successfully tracks the desired object.

First an study of related work is done in which the existing techniques are analyzed. Then the work is focused on the description of the proposed approach whose main aim is to detect the frames of change in which the tracker loses or recovers the target. For this purpose a set of features related to shape, motion and appearance of the tracked object is used. When these features have sudden variations the existence of a frame of change is considered. To identify these atypical values of the features we propose an anomaly detection strategy. Then a state machine is used to decide in each frame whether tracking is correct or wrong (reliability estimated).

Finally the proposed approach is tested with six different video trackers and compared against the most relevant techniques of the state-of-the-art. For such evaluation task we have selected a dataset that includes the most common problems in video tracking.

Keywords

Video object tracking, estimation of the reliability, online evaluation, frames of change, features, anomaly detection.

Agradecimientos

En primer lugar quiero agradecer a mi tutor Juan Carlos por la ayuda que me ha proporcionado para la realización de este proyecto.

También doy las gracias a mis padres y a mi hermano Carlos por todo su apoyo a lo largo la carrera.

Álvaro Calvo Tapia
Madrid, julio de 2014

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	3
1.3. Organización de la memoria	4
2. Estado del arte	5
2.1. Clasificación de los métodos de estimación de fiabilidad	5
2.2. Técnicas de estimación de fiabilidad seleccionadas	7
2.2.1. Clasificación CT/WT con <i>Naive Bayes</i>	7
2.2.2. <i>Backtracking</i> utilizando cadenas de Markov	8
2.2.3. ARTE (<i>Adaptive Reverse Tracking Evaluation</i>)	10
2.3. Consideraciones finales	13
3. Algoritmo	15
3.1. Estructura del algoritmo	15
3.2. Detección de anomalías	19
3.2.1. Versiones propuestas	19
4. Características	23
4.1. Forma	23
4.1.1. Variación de la relación de aspecto	23
4.1.2. Variación del área	24
4.2. Movimiento	24
4.2.1. Suavidad en la dirección	24
4.2.2. Suavidad en el desplazamiento	25
4.3. Apariencia	26
4.3.1. Diferencia entre los histogramas	26
4.3.2. SSD (<i>Sum of Squared Differences</i>)	27
4.3.3. Diferencia de textura	27

4.3.4.	SSIM (<i>Structural SIMilarity</i>)	29
4.3.5.	Diferencia entre los HOG (<i>Histograms of Oriented Gradients</i>)	31
4.3.6.	Diferencia entre los CLD (<i>Color Layout Descriptors</i>)	32
4.3.7.	Diferencia entre las matrices de covarianzas	34
4.4.	Resumen	36
5.	Resultados experimentales	37
5.1.	Implementación de las características y del algoritmo	37
5.2.	<i>Trackers</i>	37
5.3.	<i>Dataset</i>	38
5.4.	Métricas	40
5.5.	Análisis de las características	41
5.5.1.	Modelado	42
5.5.2.	Correlación	45
5.6.	Evaluación del algoritmo propuesto	47
5.6.1.	Evaluación usando detección de anomalías	47
5.6.2.	Evaluación usando clasificación supervisada	55
5.7.	Comparativa con el estado del arte	56
6.	Conclusiones y trabajo futuro	61
6.1.	Resumen	61
6.2.	Conclusiones	62
6.3.	Trabajo futuro	62
	Bibliografía	65
	Apéndice.	73
A.	Otras estrategias para la detección de los <i>frames</i> de cambio	73
A.1.	Clasificación supervisada	73
A.1.1.	Evaluación del algoritmo propuesto	74
B.	SOVTds (<i>Single Object Video Tracking dataset</i>)	79
B.1.	Factores de complejidad	79
B.2.	Situaciones modeladas	80
C.	Presupuesto	81
D.	Pliego de condiciones	83

Índice de figuras

1.1. Ejemplos de distintos problemas que aparecen en escenarios reales	2
1.2. Ejemplos de seguimiento correcto y de seguimiento incorrecto	2
2.1. Diagrama de bloques para la clasificación CT/WT con <i>Naive Bayes</i>	8
2.2. Esquema del algoritmo de <i>backtracking</i> utilizando cadenas de Markov	9
2.3. Esquema de la aproximación rápida del algoritmo de <i>backtracking</i>	10
2.4. Diagrama de bloques de ARTE	11
2.5. Máquina de estados de ARTE para determinar la condición del <i>tracker</i>	12
2.6. Máquina de estados de ARTE para determinar la segmentación temporal	13
3.1. Diagrama de bloques del algoritmo propuesto	15
3.2. Máquina de estados para la decisión CT/WT en cada <i>frame</i>	17
3.3. Ejemplo de secuencia con <i>frames</i> de cambio CT→WT y WT→CT	18
3.4. Diagrama de bloques de la técnica de detección de anomalías	20
4.1. Ejemplo de uso de las características relacionadas con la forma	24
4.2. Ejemplo de uso de las características relacionadas con el movimiento	25
4.3. Ejemplo de uso de las características relacionadas con la apariencia	26
4.4. Banco de filtros de Gabor con diferentes escalas y orientaciones	28
4.5. Diagrama de bloques del método SSIM	29
4.6. Ejemplo de comparación de imágenes utilizando MSSIM	31
4.7. Proceso de cálculo de los descriptores HOG	32
4.8. Proceso de extracción del CLD	33
5.1. Ejemplos de <i>frames</i> pertenecientes a secuencias de cada uno de los niveles del <i>dataset</i> SOVTds	39
5.2. Ejemplo de solape espacial entre la imagen del objeto estimada por el <i>tracker</i> y la imagen obtenida a partir de las anotaciones de GT	41
5.3. Ejemplo de curva ROC	41

5.4. Ejemplo de funcionamiento del algoritmo propuesto utilizando <i>covariance matrix difference</i> sobre MS	49
5.5. Ejemplo de funcionamiento del algoritmo propuesto utilizando <i>HOG difference</i> sobre MS	50
5.6. Ejemplo de funcionamiento del algoritmo propuesto utilizando <i>CLD difference</i> sobre CPF	51
5.7. Ejemplo de funcionamiento del algoritmo propuesto utilizando <i>shape ratio score</i> sobre LOT	52
5.8. Ejemplo de funcionamiento del algoritmo propuesto utilizando <i>texture difference</i> sobre TLD	53
5.9. Curvas ROC para la segmentación temporal CT/WT utilizando el algoritmo propuesto y dos técnicas del estado del arte: clasificación CT/WT con <i>Naive Bayes</i> y <i>backtracking</i>	59
5.10. Coste computacional (ms/frame) frente a AUC para el algoritmo propuesto y para dos técnicas del estado del arte: clasificación CT/WT con <i>Naive Bayes</i> y <i>backtracking</i>	60
A.1. Diagrama de bloques genérico de las técnicas de clasificación supervisada	74

Índice de tablas

2.1. Clasificación de los métodos de estimación de fiabilidad	7
4.1. Pesos utilizados para los coeficientes del descriptor CLD	34
4.2. Resumen de los aspectos más importantes de las características seleccionadas . .	36
5.1. Niveles del <i>dataset</i> SOVTds	39
5.2. Distancia de Bhattacharyya entre los histogramas de los distintos modelos de CT y el histograma del modelo de WT	43
5.3. Distancia de Bhattacharyya entre los histogramas de los modelos de CT obtenidos a partir de los datos de seguimiento de todos los <i>trackers</i> y el histograma del modelo de CT obtenido a partir de los datos de GT.	43
5.4. Estadísticos obtenidos en el test de Kolmogórov-Smirnov	44
5.5. Coeficiente de correlación de Pearson (en valor absoluto) entre cada una de las características	46
5.6. Resultados obtenidos con el algoritmo propuesto usando <i>un modelo de CT por cada característica</i>	48
5.7. Resultados obtenidos con el algoritmo propuesto usando <i>un único modelo de CT multivariante</i>	54
5.8. Resultados obtenidos con el algoritmo propuesto <i>combinando las detecciones de las características</i>	55
5.9. Valor medio de los resultados obtenidos con el algoritmo propuesto utilizando clasificadores globales	56
5.10. Valor medio de los resultados obtenidos con el algoritmo propuesto utilizando métodos de combinación de clasificadores débiles	56
5.11. Comparativa de los resultados conseguidos con el algoritmo propuesto y los obte- nidos con dos técnicas del estado del arte: clasificación CT/WT con <i>Naive Bayes</i> y <i>backtracking</i>	58
5.12. Comparativa del coste computacional (ms/ <i>frame</i>) del algoritmo propuesto y de dos técnicas del estado del arte: clasificación CT/WT con <i>Naive Bayes</i> y <i>backtracking</i>	60

A.1. Resultados obtenidos con el algoritmo propuesto utilizando clasificadores globales	76
A.2. Resultados obtenidos con el algoritmo propuesto utilizando métodos de combinación de clasificadores débiles	77
B.1. Criterios y factores para estimar la complejidad de los problemas de seguimiento	79

Acrónimos

CT	<i>Correct Tracking</i>
WT	<i>Wrong Tracking</i>
GT	<i>Ground-Truth</i>
ARTE	<i>Adaptive Reverse Tracking Evaluation</i>
SSIM	<i>Structural SIMilarity</i>
DSSIM	<i>Structural DisSIMilarity</i>
HOG	<i>Histogram of Oriented Gradients</i>
CLD	<i>Color Layout Descriptor</i>
SSD	<i>Sum of Squared Differences</i>
RGB	<i>Red Green Blue color model</i>
DCT	<i>Discrete Cosine Transform</i>
SVM	<i>Support Vector Machine</i>
KNN	<i>K-Nearest Neighbors</i>
MS	<i>Mean Shift</i>
CPF	<i>Color-based Particle Filter</i>
LOT	<i>Locally Orderless Tracking</i>
IVT	<i>Incremental Visual Tracking</i>
SPT	<i>SuperPixel Tracking</i>
TLD	<i>Tracking-Learning-Detection</i>

TP	<i>True Positive</i>
FP	<i>False Positive</i>
TN	<i>True Negative</i>
FN	<i>False Negative</i>
TPR	<i>True Positive Rate</i>
FPR	<i>False Positive Rate</i>
AUC	<i>Area Under the Curve</i>
ROC	<i>Receiver Operating Characteristic</i>

Capítulo 1

Introducción

En este capítulo se ofrece una introducción al trabajo presentado en este documento. En las siguientes secciones se describe la motivación del proyecto (Sección 1.1), los objetivos principales (Sección 1.2) y la organización de la memoria (Sección 1.3).

1.1. Motivación

En la actualidad el seguimiento de objetos en vídeo (*tracking*) es una etapa muy importante en diversas áreas como vídeo-vigilancia, interacción persona-ordenador, monitorización de tráfico, indexación de vídeo, compresión de vídeo basada en objetos, robótica e imágenes médicas. Debido al gran interés existente, a día de hoy se dispone de una gran cantidad de algoritmos de seguimiento (*trackers*) [1, 2, 3]. Para llevar a cabo la selección del algoritmo óptimo en cada aplicación, normalmente se emplea la experiencia del operador humano. Se trata de una tarea muy difícil, ya que los datos de vídeo que son analizados presentan múltiples problemas tales como iluminación variable, ruido, cambios de pose, cambios de escala, oclusiones y partes de la imagen similares al objeto seguido (ver Figura 1.1). Bajo estas condiciones ningún algoritmo podrá funcionar perfectamente en todas las situaciones, por lo que se esperan fallos cuando se trabaje con escenarios reales (ver Figura 1.2).

Para evaluar la robustez de los algoritmos de seguimiento la solución clásica se basa en comparar sus resultados frente a anotaciones manuales de las posiciones correctas del objeto (*ground-truth*) [1, 4, 5, 6, 7]. Esta evaluación es muy limitada pues la anotación es una actividad tediosa sujeta a errores humanos. Además sólo se utiliza un escaso número de secuencias de test que no cubren con suficiente variabilidad todas las situaciones de interés y sus problemas asociados. Por tanto es difícil extrapolar los resultados de la evaluación a nuevas secuencias y las conclusiones que se pueden extraer no son claras.

Puesto que la evaluación basada en *ground-truth* es altamente costosa, ha surgido la necesidad de explorar otras alternativas. Recientemente se ha propuesto la estimación de fiabilidad [8, 9,



Figura 1.1: Ejemplos de distintos problemas que aparecen en escenarios reales. El rectángulo verde indica el objeto seguido. De arriba abajo y de izquierda a derecha: iluminación variable, ruido, cambio de pose, cambio de escala, oclusión y partes de la imagen similares al objeto seguido (rectángulos rojos).

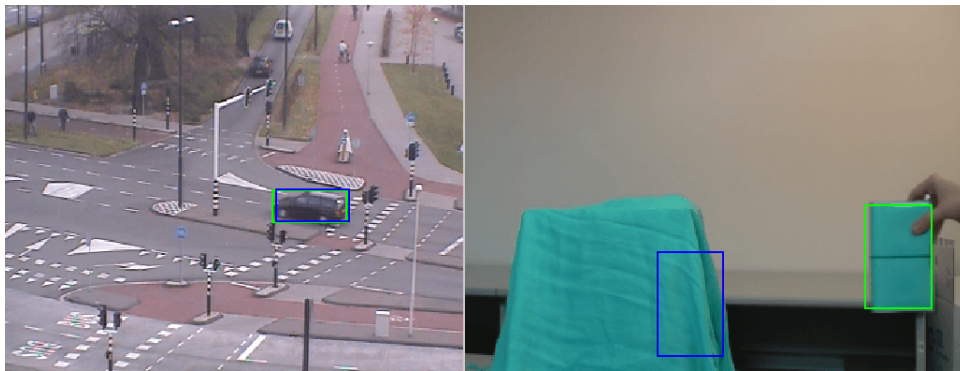


Figura 1.2: Ejemplos de seguimiento correcto (izquierda) y de seguimiento incorrecto (derecha). El rectángulo azul marca el área estimada por el algoritmo de seguimiento (*tracker*) y el rectángulo verde el área anotada manualmente (*ground-truth*).

10, 11, 12] como tarea equivalente, que persigue medir la calidad de los *trackers* en tiempo de ejecución (*online*) utilizando información extraída a partir de los datos de seguimiento y en ausencia de *ground-truth*. Las ventajas de este tipo de análisis son múltiples; entre otras, la evaluación sobre grandes *datasets* sin anotar, auto-ajuste, rankings comparativos y combinación de algoritmos.

Los métodos de estimación de fiabilidad existentes en la actualidad se basan en propiedades de las trayectorias estimadas del objeto, tales como suavidad en el movimiento, consistencia del área o reversibilidad temporal; en propiedades estadísticas de la salida del *tracker*, tales como la probabilidad de observación (*likelihood*), incertidumbre espacial o comprobaciones de la consistencia; en características complementarias, tales como contraste de color o potencia discriminativa de fondo; y en combinación de estas propiedades. Sin embargo estos métodos la mayoría de las veces son dependientes de la aplicación, no se pueden utilizar con secuencias largas o no son adaptables a pérdidas y recuperaciones del objeto por parte del *tracker*. Por tanto su uso se restringe normalmente a vídeos con pocos *frames* o de baja complejidad. La mejora del estado del arte supone entonces un reto. Por este motivo mediante el presente trabajo se pretende contribuir afrontando las citadas limitaciones.

1.2. Objetivos

El principal objetivo de este Proyecto Fin de Carrera (PFC) es el diseño, implementación y evaluación de una técnica de estimación de fiabilidad de algoritmos de seguimiento de objetos en vídeo. Este objetivo general se desglosa en los siguientes objetivos más concretos:

- Estudio del estado del arte actual en el que se analizarán algunas propuestas de estimación de fiabilidad existentes.
- Diseño e implementación de una nueva técnica para estimar la fiabilidad de algoritmos de seguimiento de objetos. Para ello se explorarán diferentes características relativas a la forma, al movimiento y a la apariencia del objeto.
- Selección de una base datos que contenga secuencias de vídeo sintéticas y reales que representen los distintos problemas existentes (oclusiones, cambios de iluminación, ruido, objetos similares, cambios de escala...). Para ello se utilizará material *online* disponible [13].
- Evaluación de la técnica implementada. Se hará especial hincapié en la estimación de fiabilidad sobre diversos algoritmos de seguimiento disponibles en el Grupo de Tratamiento e Interpretación de Vídeo de la Universidad Autónoma de Madrid (VPU-Lab).

1.3. Organización de la memoria

La memoria consta de los siguientes capítulos:

- Capítulo 1. *Introducción*: Introducción, motivación del proyecto y objetivos.
- Capítulo 2. *Estado del arte*: Estudio del estado del arte de la estimación de fiabilidad.
- Capítulo 3. *Algoritmo*: Descripción del algoritmo propuesto.
- Capítulo 4. *Características*: Presentación de las características seleccionadas.
- Capítulo 5. *Resultados experimentales*: Exposición de los resultados obtenidos con el algoritmo propuesto.
- Capítulo 6. *Conclusiones y trabajo futuro*: Resumen del proyecto, conclusiones finales y futuras líneas de investigación

Al final del documento se incluyen además varios apéndices:

- Apéndice A. *Otras estrategias para la detección de los frames de cambio*.
- Apéndice B. *SOVTds (Single Object Video Tracking dataset)*.
- Apéndice C. *Presupuesto*.
- Apéndice D. *Pliego de condiciones*.

Capítulo 2

Estado del arte

En este capítulo se estudia el estado del arte relacionado con la estimación de fiabilidad del seguimiento de objetos en vídeo. Primero se lleva a cabo una clasificación de los métodos de estimación de fiabilidad existentes (Sección 2.1). Después se analizan varias técnicas actuales (Sección 2.2). Para finalizar se incluyen algunas consideraciones que han servido como punto de partida para el desarrollo del algoritmo propuesto en este trabajo (Sección 2.3).

2.1. Clasificación de los métodos de estimación de fiabilidad

Los métodos tradicionales para evaluar la robustez de los algoritmos de seguimiento de objetos en vídeo se basan en la información de *ground-truth*. La generación de estas anotaciones manuales es una tarea tediosa que consume mucho tiempo y que, además, no está exenta de posibles errores. Por ello en la actualidad se está investigando como alternativa la estimación de fiabilidad, cuyo objetivo es medir la calidad de los *trackers* en tiempo de ejecución (*online*) a partir de los datos de seguimiento y en ausencia de *ground-truth*. Los métodos de estimación de fiabilidad propuestos en la literatura se pueden clasificar en tres categorías principales [8]:

- **Métodos basados en la trayectoria.** Estos métodos utilizan las posiciones estimadas del objeto a lo largo del tiempo. Se identifican tres subcategorías:
 - **Métodos *forward*.** Estos métodos analizan la trayectoria en ventanas temporales cortas. Emplean características que miden, por ejemplo, la longitud de la trayectoria [14] o la suavidad del desplazamiento [14, 15, 16] o de la dirección [15, 17, 18]. En general son métodos que proporcionan sólo una decisión binaria y son dependientes de la aplicación.
 - **Métodos *backward*.** Estos métodos hacen uso de la propiedad de reversibilidad temporal que asume que el movimiento de un objeto en el tiempo también se produce

en el sentido contrario y, por tanto, el *tracker* debería ser capaz de seguir al objeto en la dirección inversa [12, 19, 20]. Son preferibles a otros métodos basados en la trayectoria, aunque si el análisis hacia atrás es corto sufren acumulación de errores y si es largo son computacionalmente irrealizables. Por tanto su aplicación está restringida a secuencias con pocos *frames*.

- **Métodos basados en modelos.** Estos métodos miden la similitud entre las nuevas trayectorias del objeto y modelos de trayectoria calculados *online* [21, 22]. Su aplicación es limitada ya que necesitan una gran cantidad de datos para entrenar los modelos.
- **Métodos basados en características.** Estos métodos analizan la salida del algoritmo de seguimiento o sus etapas internas. Se distinguen dos subcategorías:
 - **Métodos basados en la diferencia de las características.** Estos métodos miden variaciones de las características relacionadas con las diferencias de color entre el frente y el fondo [23, 24] o con el contraste a lo largo del contorno del objeto [25, 26]. Son métodos dependientes de la aplicación puesto que estas variaciones no se pueden garantizar en todos los escenarios (por ejemplo, cuando el objeto es similar al fondo).
 - **Métodos basados en la consistencia de las características.** Estos métodos se fundamentan en estadísticos para comprobar la consistencia en el tiempo de la forma [27], la escala [15] o la apariencia [17, 25, 28, 29] del objeto. Se aplican a *trackers* probabilísticos aprovechando que éstos proporcionan una estimación del estado del objeto que se utiliza para calcular estadísticos relacionados con la probabilidad de observación [30, 31, 32], la covarianza del estado del objeto [30, 33, 34] o tests como χ^2 [35] y Kolmogórov-Smirnov [36]. Aunque estos métodos producen mejoras, fallan cuando el objeto se mueve por zonas con diferentes niveles de ruido o cuando el *tracker* sigue a otro objeto similar, ya que en estos casos puede verse afectada la probabilidad de observación (*likelihood*).
- **Métodos híbridos.** Estos métodos son combinaciones de los anteriores. Por ejemplo, en [19] y [37] se utilizan conjuntamente reversibilidad temporal con métodos basados en diferencia de las características; en [17] se emplea suavidad en la dirección y en el movimiento con consistencia del color; en [14] y [15] se combinan múltiples características tales como suavidad en el movimiento, complejidad de la trayectoria o consistencia de la forma y del color.

Aunque existen propuestas variadas en cada categoría, la mayor parte presentan limitaciones. En general casi todos los métodos dependen de la aplicación ya que emplean un conocimiento

a priori del algoritmo a ser evaluado. Además muchos de ellos identifican empíricamente las características que usan y su contribución a la evaluación. En algunos casos, como los métodos *backward*, su aplicabilidad queda reducida a secuencias cortas debido a su elevado coste computacional. Asimismo hay métodos basados en características que fallan puesto que no son capaces de adaptarse a determinados errores del *tracker*.

Para terminar esta sección se muestra en la Tabla 2.1 un resumen de las categorías descritas.

Categoría	Subcategoría	Características	Métricas	Trackers
Trayectoria	<i>Forward</i>	Tamaño y posición	Euclídea	D & P
	<i>Backward</i>	Posición y modelos de espacio de estado	Mahalanobis y Euclídea	D & P
	Modelos	Posición	Euclídea	D & P
Características	Diferencia	Posición y contorno	Bhattacharyya y Euclídea	D & P
	Consistencia	Tamaño, apariencia y modelos de espacio de estado	Teoría de la información y detección de cambios	P
Híbridos	-	Tamaño, posición y apariencia	Euclídea	D & P

Tabla 2.1: Clasificación de los métodos de estimación de fiabilidad (extraída de [8]). (*Trackers*: D - Determinístico, P - Probabilístico)

2.2. Técnicas de estimación de fiabilidad seleccionadas

En los siguientes apartados se realiza un análisis de tres técnicas de estimación de fiabilidad recientes: clasificación CT/WT con *Naive Bayes* (Sección 2.2.1), *backtracking* utilizando cadenas de Markov (Sección 2.2.2) y ARTE (Sección 2.2.3).

2.2.1. Clasificación CT/WT con *Naive Bayes*

Esta técnica [11] propone calcular un conjunto de características¹ relacionadas con la forma, el movimiento y la apariencia del objeto seguido y combinarlas para decidir en cada *frame* si el seguimiento es correcto (CT) o incorrecto (WT) mediante el uso de un clasificador *Naive Bayes* [38]. Este clasificador se fundamenta en el teorema de Bayes para estimar la probabilidad a posteriori de que un vector de características pertenezca a cierta clase, dadas las distribuciones estimadas de cada característica para dicha clase. Como hipótesis el clasificador asume independencia condicional entre las características utilizadas en cada clase.

Para cada característica se definen dos clases: CT y WT. En primer lugar se entrena al clasificador con vectores de características a los que se les asigna una etiqueta de clase a partir de las anotaciones de *ground truth* (GT). Posteriormente el clasificador procesa los vectores de

¹Estas características (o adaptaciones de las mismas) se utilizan junto con otras en el algoritmo propuesto en este trabajo. En el Capítulo 4 se ofrece una descripción detallada de todas ellas.

características de las secuencias de test y determina a cuál de las dos clases pertenece. En la Figura 2.1 se muestra un esquema del algoritmo.

Como se verá más adelante el problema de esta técnica es que las características que se plantean se comportan de igual manera tanto para CT como para WT y, por tanto, el clasificador no es capaz de diferenciar con claridad ambas clases.

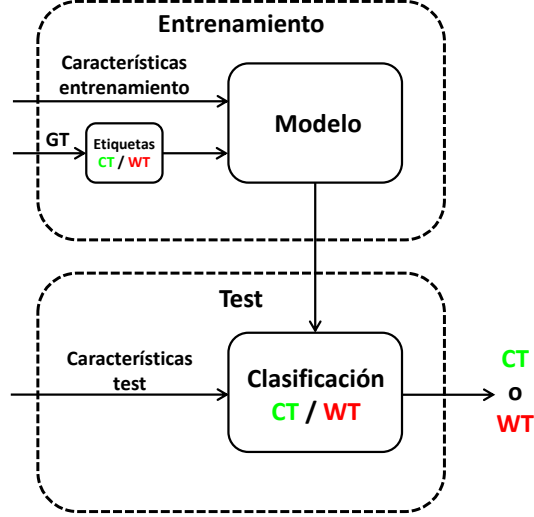


Figura 2.1: Diagrama de bloques para la clasificación CT/WT con *Naive Bayes*.

2.2.2. *Backtracking* utilizando cadenas de Markov

Esta técnica [12] se centra en sistemas basados en el filtro de partículas [39]. La idea es emplear una cadena de Markov en sentido inverso aprovechando la reversibilidad del movimiento del objeto.

El filtro de partículas sigue un esquema que representa una cadena de Markov, donde en cada instante t el estado \mathbf{x}_t se estima a partir de las observaciones $\{\mathbf{y}_1, \dots, \mathbf{y}_t\}$. Los modelos que definen el sistema (*forward*) son los siguientes:

$$\text{Cadena de Markov } \begin{matrix} \text{forward} \end{matrix} \left\{ \begin{array}{ll} \text{Densidad a priori en } t = 0: & p(\mathbf{x}_0) \\ \text{Modelo de transición:} & p(\mathbf{x}_t | \mathbf{x}_{t-1}) \\ \text{Modelo de observación:} & p(\mathbf{y}_t | \mathbf{x}_t) \end{array} \right. \quad (2.1)$$

Dada una secuencia de observaciones $\mathbf{Y}_t = \{\mathbf{y}_1, \dots, \mathbf{y}_t\}$, la función de densidad a posteriori será $\pi_t = p(\mathbf{x}_t | \mathbf{Y}_t)$. Esta función es aproximada por el filtro mediante un conjunto $S_t = \{\mathbf{x}_t^{(n)}, w_t^{(n)}\}_{n=1, \dots, N}$ de N partículas ponderadas, donde $\mathbf{x}_t^{(n)}$ representa un hipotético estado del

objeto y $w_t^{(n)}$ es el peso asociado. Cada conjunto S_t se calcula recursivamente a partir del anterior S_{t-1} . El conjunto inicial S_0 se obtiene muestreando la función de densidad a priori $p(\mathbf{x}_0)$.

Para estimar la fiabilidad del sistema (*forward*) se propone utilizar una cadena de Markov en sentido contrario que utilice π_t como su densidad a priori y la secuencia de observaciones \mathbf{Y}_t en orden invertido ($\mathbf{Y}_t^b = \{\mathbf{y}_t, \dots, \mathbf{y}_1\}$). Para llevar a cabo la evaluación en un determinado instante $t = T$ el sistema *backward* se inicializa en $t = T + 1$. Usando ahora la notación $q(\cdot)$ para las funciones de densidad de probabilidad, dicho sistema queda caracterizado como sigue:

$$\text{Cadena de Markov } \begin{cases} \text{Densidad a priori en } t = T + 1: & q(\mathbf{x}_{T+1}) = p(\mathbf{x}_{T+1} | \mathbf{Y}_T) = \\ & = \int p(\mathbf{x}_{T+1} | \mathbf{x}_T) p(\mathbf{x}_T | \mathbf{Y}_T) d\mathbf{x}_T \\ \text{Modelo de transición:} & q(\mathbf{x}_t | \mathbf{x}_{t+1}) = \frac{p(\mathbf{x}_{t+1} | \mathbf{x}_t) p(\mathbf{x}_t)}{p(\mathbf{x}_{t+1})}, \quad t \in (0, T) \\ \text{Modelo de observación:} & q(\mathbf{y}_t | \mathbf{x}_t) = p(\mathbf{y}_t | \mathbf{x}_t), \quad \forall t \end{cases} \quad (2.2)$$

Si en el sistema *backward* definido se filtra la secuencia de observaciones en sentido opuesto ($\mathbf{Y}_T^b = \{\mathbf{y}_T, \dots, \mathbf{y}_1\}$), la función de densidad a posteriori para cada instante t será en esta ocasión $\pi_t^b = q(\mathbf{x}_t | \mathbf{Y}_t^b) = q(\mathbf{x}_t | \mathbf{y}_T, \mathbf{y}_{T-1}, \dots, \mathbf{y}_t)$. Aplicando la recursión se puede llegar a obtener en el punto de referencia $t = 0$ la densidad π_0^b (ver Figura 2.2).

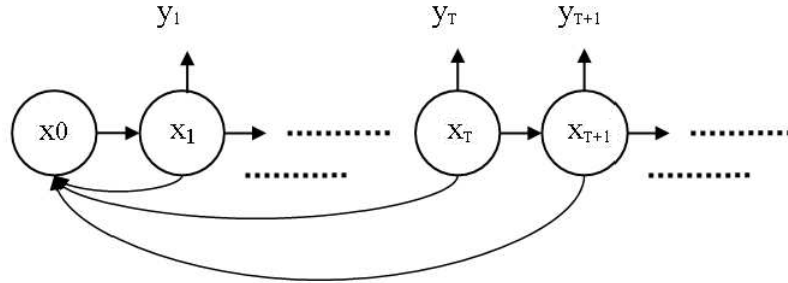


Figura 2.2: Esquema del algoritmo de *backtracking* utilizando cadenas de Markov (extraído de [12]). Se toma como punto de referencia el instante $t = 0$.

Si en el instante $t = T$ el sistema *forward* tiene suficiente información del objeto, entonces el sistema *backward* será capaz de seguirlo hasta el *frame* inicial. En ese caso se espera que la densidad a posteriori π_0^b obtenida con el sistema *backward* sea parecida, desde un punto de vista estadístico, a la densidad a priori $p(\mathbf{x}_0)$ del sistema *forward*. Para medir esa similitud se usa la distancia de Mahalanobis [40] entre la distribución a priori p y la distribución a posteriori π :

$$d(p, \pi) = (\boldsymbol{\mu}_p - \boldsymbol{\mu}_\pi)^T \boldsymbol{\Sigma}_p^{-1} (\boldsymbol{\mu}_p - \boldsymbol{\mu}_\pi) + (\boldsymbol{\mu}_p - \boldsymbol{\mu}_\pi)^T \boldsymbol{\Sigma}_\pi^{-1} (\boldsymbol{\mu}_p - \boldsymbol{\mu}_\pi), \quad (2.3)$$

donde $\boldsymbol{\mu}_p$ y $\boldsymbol{\Sigma}_p$ son, respectivamente, el vector de medias y la matriz de covarianzas de la

distribución p y μ_π y Σ_π indican lo mismo referido a la distribución π .

Aunque la técnica está diseñada para sistemas que se fundamentan en el filtro de partículas, los autores sugieren que la idea se puede extender a otro tipo de *trackers*. En este caso sería necesaria una adaptación para poder aplicarla.

La principal desventaja de este algoritmo es que su coste computacional crece linealmente con el número de *frames*. Por tanto puede llegar a hacerse inviable si las secuencias son muy largas. Para hacer frente a este inconveniente se plantea una aproximación rápida en la cual el punto de referencia se va desplazando del *frame* inicial a *frames* en los cuales se asume que el seguimiento es correcto. Se crean así múltiples puntos de referencia separados Δt *frames* (ver Figura 2.3). De esta manera dado un instante $t = T$ su punto de referencia será $t = \Delta t \left\lfloor \frac{T}{\Delta t} \right\rfloor$. Escogiendo un valor adecuado de Δt deberá buscarse entonces un compromiso entre el tiempo de ejecución, que será proporcional a Δt , y la capacidad de detectar pequeños cambios en Δt .

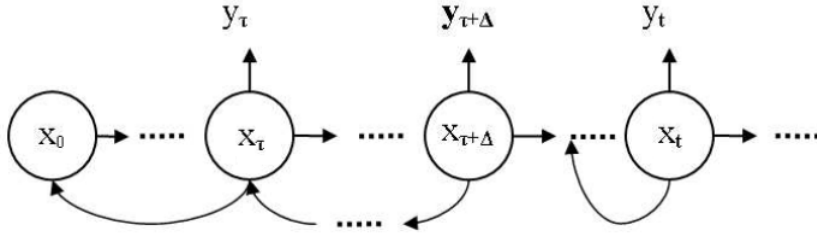


Figura 2.3: Esquema de la aproximación rápida del algoritmo de *backtracking* (extraído de [12]). Se crean múltiples puntos de referencia separados Δt *frames*.

2.2.3. ARTE (*Adaptive Reverse Tracking Evaluation*)

Esta técnica [8] combina un análisis de incertidumbre del *tracker* con la propiedad de reversibilidad temporal. Su funcionamiento se demuestra sobre el filtro de partículas de color (CPF) [41]. Se divide en dos etapas: en la primera se estima la condición del *tracker* para identificar los segmentos temporales en los que el objeto se pierde; en la segunda se mide la calidad de seguimiento en los instantes en los que el *tracker* sigue al objeto correcto. En la Figura 2.4 se muestra el diagrama de bloques de esta técnica.

Para describir la condición del *tracker* se definen tres posibles eventos: *locked-on*, cuando el *tracker* está siguiendo a un objeto que puede ser el correcto u otro de similares características; *locking-in*, cuando el *tracker* está recuperándose de un error y comienza a seguir a un objeto; y *scanning*, cuando el *tracker* está buscando al objeto después de un fallo. Analizando la salida del filtro en cada instante se determina la incertidumbre del *tracker* U_t , la cual proporcionará información acerca de la condición.

Cuando el seguimiento sea correcto se espera que el valor de la incertidumbre sea constante

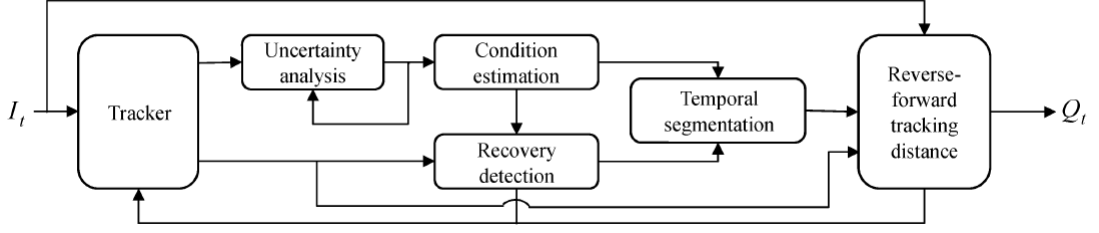


Figura 2.4: Diagrama de bloques de ARTE (extraído de [8]).

o ligeramente decreciente. Si el *tracker* pierde al objeto entonces ese valor aumentará. Tras un fallo, si el valor de la incertidumbre disminuye, indicará que el *tracker* ha empezado a seguir al objeto correcto o a otro similar. El objetivo es detectar esos cambios temporales en los niveles de la incertidumbre, los cuales permitirán determinar la condición del *tracker*. Para ello se define la señal de cambio $C_t^{W,k}$, que maximiza la diferencia entre la incertidumbre del filtro en el *frame* actual t y valores previos de la incertidumbre dentro de una ventana temporal W :

$$C_t^{W,k} = \frac{U_t - U_{\hat{t}}}{U_k}, \quad (2.4)$$

donde:

$$\hat{t} = \arg \max_{j \in W} \left(\left| \frac{U_t - U_j}{U_k} \right| \right) \quad (2.5)$$

y $k \in \{\hat{t}, t\}$, con $k = \hat{t}$ para detectar las transiciones crecientes de la incertidumbre y $k = t$ para detectar las transiciones decrecientes. Los cambios lentos se detectan utilizando una ventana temporal larga W_1 mientras que los cambios repentinos se detectan empleando una ventana temporal corta W_2 . Esto da lugar cuatro señales de cambio: $C_t^{W_1, \hat{t}}$, $C_t^{W_2, \hat{t}}$, $C_t^{W_1, t}$ y $C_t^{W_2, t}$.

Para decidir la condición del *tracker* en cada *frame* se usa la máquina de estados de la Figura 2.5. Las transiciones entre los estados vienen dadas por los cambios globales y locales detectados. Las condiciones de los cambios globales GI_t y GD_t se definen como:

$$GI_t = \begin{cases} 1 & \text{si } C_t^{W_1, \hat{t}} \geq \tau_1 \vee C_t^{W_2, \hat{t}} \geq \tau_1 \\ 0 & \text{resto} \end{cases}, \quad (2.6)$$

$$GD_t = \begin{cases} 1 & \text{si } C_t^{W_1, t} \geq \tau_2 \vee C_t^{W_2, t} \geq \tau_2 \\ 0 & \text{resto} \end{cases}, \quad (2.7)$$

donde τ_i ($i = 1, 2$) es el cambio relativo (por ejemplo, $\tau_i = 2$ significa un 200 % de cambio).

La máquina de estados parte de *locked-on* cuando el *tracker* se inicializa. Si se detecta un

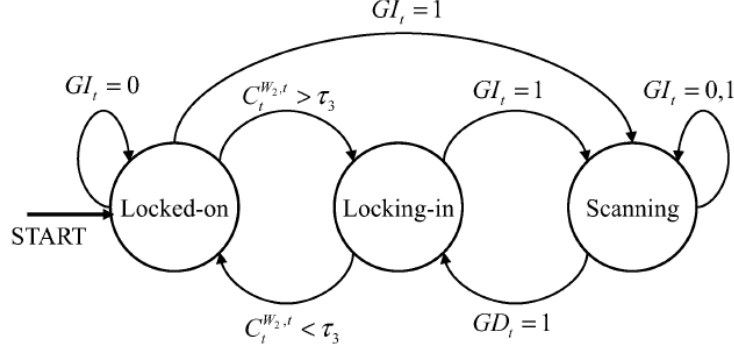


Figura 2.5: Máquina de estados de ARTE para determinar la condición del *tracker* (extraída de [8]).

incremento global de la incertidumbre ($GI_t = 1$) pasa a *scanning* y ante un cambio repentino decreciente ($C_t^{W_2,t} > \tau_3$) pasa a *locking-in*. τ_3 evalúa cuánto ha decrecido la señal en el cambio (por ejemplo, $\tau_3 = \tau_2/2$). En el estado *scanning* pasa a *locking-in* si hay un decrecimiento global de la incertidumbre ($GD_t = 1$). Desde ese estado pasa a *locked on* en caso de que la incertidumbre se estabilice ($C_t^{W_2,t} < \tau_3$) o vuelve a *scanning* si hay un incremento global ($GI_t = 1$).

El análisis de la incertidumbre por sí solo no puede determinar si el *tracker* ha recuperado al objeto tras un fallo ya que en el estado *locked-on* puede ocurrir que el objeto seguido sea otro de similares características. Es decir, en esa situación el nivel de incertidumbre puede ser el mismo que cuando el *tracker* sigue al objeto correcto. Para afrontar esta limitación se propone utilizar la propiedad de reversibilidad temporal cada vez que haya una transición de *scanning* a *locked on* a través de *locking-in*. Por tanto en esos casos se lleva a cabo un análisis *backward* desde el instante actual t hasta el último instante t_0 en el que el *tracker* (*forward tracker*) estimó correctamente el estado del objeto antes de perderlo. Para comprobar si verdaderamente se ha recuperado al objeto se mide en t_0 el solape espacial entre la imagen del estado $\mathbf{x}_{t_0}^F$ estimado por el *tracker* (*forward tracker*) y la imagen del estado $\mathbf{x}_{t_0}^R$ estimado por *tracker* inverso (*reverse tracker*). Se usa para ello el coeficiente de Dice que se define como:

$$d_S(\mathbf{x}_{t_0}^F, \mathbf{x}_{t_0}^R) = \frac{2 |A_{t_0}^F \cap A_{t_0}^R|}{|A_{t_0}^F| + |A_{t_0}^R|}, \quad (2.8)$$

donde $|A_{t_0}^F \cap A_{t_0}^R|$ es el solape espacial (en píxeles) de las imágenes y $|A_{t_0}^F|$ y $|A_{t_0}^R|$ representan sus respectivas áreas (en píxeles). Si el valor obtenido supera un determinado umbral (τ_4) entonces se considerará que el *tracker* ha recuperado al objeto.

Por último, para determinar los segmentos temporales en los que el seguimiento es correcto/incorrecto, se utiliza la máquina de estados de la Figura 2.6. Las transiciones entre ambos estados se definen en base a dos condiciones: H_1 y H_2 . Asumiendo que la máquina comienza en

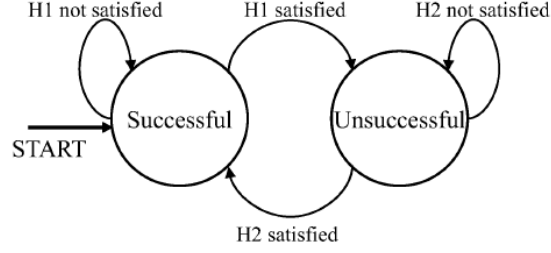


Figura 2.6: Máquina de estados de ARTE para determinar la segmentación temporal (extraída de [8]).

el estado *successful* al inicializarse el *tracker*, H_1 se satisface cuando la condición del *tracker* pasa a *scanning* o cuando permanece en dicho estado. H_2 se cumple cuando la condición del *tracker* pasa de *locking-in* a *locked-on* y además se recupera al objeto ($d_S(\mathbf{x}_{t_0}^F, \mathbf{x}_{t_0}^R) \geq \tau_4$).

Después de la segmentación temporal se estima la calidad de seguimiento en los intervalos en los que el *tracker* sigue al objeto correcto. El resto de segmentos se descartan ya que el objeto se ha perdido. Para evaluar la calidad en el *frame* t se aplica la propiedad de reversibilidad temporal y se mide en cada instante i la diferencia entre el estado \mathbf{x}_i^F estimado por el *tracker* (*forward tracker*) y el estado \mathbf{x}_i^R estimado por *tracker* inverso (*reverse tracker*).

Una de las limitaciones de esta técnica es que es aplicable únicamente a *trackers* multi-hipótesis que usen alguna forma de incertidumbre relacionada con la propagación de las hipótesis generadas. En el caso de los *trackers* de hipótesis única se requeriría una adaptación de la salida del algoritmo.

También hay que tener en cuenta que la complejidad computacional puede hacerse alta en secuencias largas en el momento de llevar a cabo el análisis *backward*.

2.3. Consideraciones finales

Como ya se ha comentado anteriormente, las características propuestas en [11] toman valores muy similares cuando el seguimiento es correcto y cuando el seguimiento es incorrecto, por lo que el clasificador *Naive Bayes* no puede distinguir claramente ambas clases. Sin embargo se ha observado que estas características suelen adquirir unos valores más atípicos en los *frames* en los que el *tracker* pierde o recupera al objeto. Se ha considerado entonces que la opción más adecuada es emplearlas para detectar esos instantes concretos. Por este motivo se ha decidido crear un nuevo algoritmo en el que se utilizan estas características (o adaptaciones de las mismas) junto con algunas más relacionadas con la apariencia del objeto.

En los siguientes capítulos se define detalladamente el algoritmo propuesto (Capítulo 3) y cada una de las características seleccionadas (Capítulo 4).

Capítulo 3

Algoritmo

En este capítulo se describe el algoritmo propuesto para la estimación de fiabilidad de los *trackers*. Su objetivo principal es identificar los *frames* en los que el *tracker* pierde o recupera al objeto utilizando para ello una estrategia de detección de anomalías. En primer lugar se proporciona una visión de la estructura del algoritmo (Sección 3.1). Posteriormente se explica el funcionamiento de la estrategia de detección de anomalías (Sección 3.2).

3.1. Estructura del algoritmo

El algoritmo que se plantea sigue un esquema formado por tres etapas (ver Figura 3.1): en la primera se extrae, a partir de los datos de seguimiento, un conjunto de características para obtener información acerca del objeto; en la segunda se emplean las características de la etapa anterior para detectar los *frames* de cambio CT→WT y WT→CT; en la tercera se lleva a cabo una segmentación temporal determinando en cada *frame* si el seguimiento es correcto (CT) o incorrecto (WT).

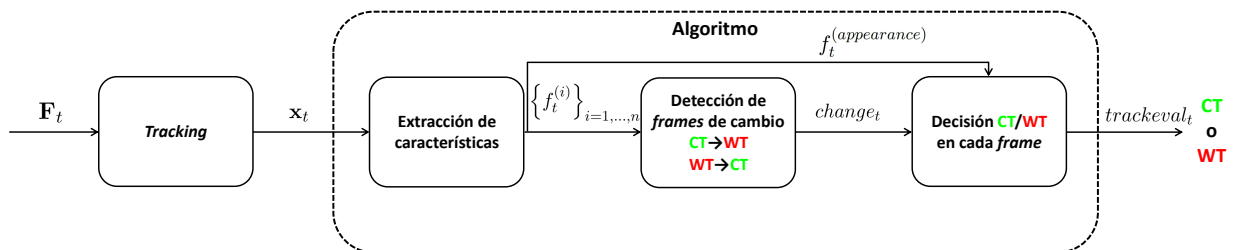


Figura 3.1: Diagrama de bloques del algoritmo propuesto.

Sea \mathbf{x}_t el estado del objeto estimado por el *tracker* en el instante t :

$$\mathbf{x}_t = f(\mathbf{F}_t, \mathbf{x}_{t-1}, \beta_{t-1}), \quad (3.1)$$

donde $f(\cdot)$ representa el algoritmo de seguimiento, \mathbf{F}_t denota el *frame* en el instante t , \mathbf{x}_{t-1} es el estado del objeto estimado en el instante $t-1$ y β_{t-1} indica el modelo del objeto¹ en el instante $t-1$. El estado \mathbf{x}_t es un vector cuyos elementos definen las coordenadas (x_t e y_t), los dos ejes (W_t y H_t) y la orientación (θ_t) del *bounding box*² que contiene al objeto seguido. Estos datos son usados por el algoritmo para estimar la fiabilidad.

A continuación se explica con detalle cada etapa:

- **Extracción de características.** En esta etapa se utiliza, en cada instante t , el vector \mathbf{x}_t para extraer un conjunto de n características $\left\{f_t^{(i)}\right\}_{i=1,\dots,n}$ relativas a la forma, al movimiento y a la apariencia del objeto. En el Capítulo 4 se especifica cada una de ellas.
- **Detección de *frames* de cambio CT→WT y WT→CT.** El objetivo de esta etapa es localizar, de manera indistinta, aquellos *frames* en los que el *tracker* pierde o recupera al objeto. Estos momentos de cambio se denominarán CT→WT y WT→CT respectivamente. Es decir, lo que se intenta buscar son los instantes t en los que la señal de error de GT³ (e_t) pasa de ser menor que uno a ser igual uno ($e_{t-1} < 1 \wedge e_t = 1$) y al contrario, los instantes t en los que pasa de ser igual a uno a ser menor que uno ($e_{t-1} = 1 \wedge e_t < 1$) (ver Figura 3.3). Para realizar estas detecciones se emplean las características extraídas en la etapa anterior, las cuales se asume que van a experimentar variaciones repentinas en los *frames* de cambio. Para discriminar esos valores atípicos de las características se propone usar una técnica de detección de anomalías (Sección 3.2). Adicionalmente se ha considerado como alternativa el uso de estrategias de clasificación supervisada (Apéndice A). Como resultado de esta etapa se obtiene la señal binaria $change_t$ que toma el valor 1 en los *frames* en los que se detecta un cambio y 0 en el resto de *frames*:

$$change_t = \begin{cases} 1 & \text{si cambio detectado} \\ 0 & \text{resto} \end{cases} \quad (3.2)$$

- **Decisión CT/WT en cada *frame*.** Una vez que se han detectado los posibles *frames* de cambio, la siguiente tarea consiste en identificar los segmentos temporales en los que el seguimiento es correcto o incorrecto (ver Figura 3.3). Es decir, el objetivo es obtener una señal binaria, la cual se denominará $trackeval_t$, que en cada instante t adquiera el valor CT

¹Si el modelo del objeto no cambia después de la inicialización del *tracker*, entonces $\beta_{t-1} = \beta_{t_0}$.

²El término *bounding box* se refiere al rectángulo o elipse que delimita el área estimada por el *tracker*.

³La señal de error de GT se define en la Fórmula 5.4.

o WT. Para ello se utiliza una sencilla máquina de dos estados⁴ (ver Figura 3.2). Partiendo inicialmente del estado CT se comprueba, en cada instante t , si se ha detectado un cambio. Mientras no exista ninguna detección ($change_t = 0$) se considerará que el objeto no se ha perdido, por lo que se seguirá en el mismo estado. En caso de haberse detectado un cambio ($change_t = 1$) se evalúa la similitud entre la imagen estimada por el *tracker* en ese *frame* y la imagen del *frame* inicial, la cual se sabe que contiene al objeto correcto⁵. El propósito de esta comparación es verificar si verdaderamente se ha perdido al objeto o si, por el contrario, se trata de una falsa detección. Para medir esa similitud se puede emplear alguna de las características relacionadas con la apariencia del objeto. Si el valor de la característica no supera el umbral ($f_t^{(appearance)} < \tau$), es decir, si el parecido es alto, entonces se considerará que la detección es errónea y se permanecerá en el estado CT. En caso contrario ($f_t^{(appearance)} > \tau$) se asumirá que el objeto se ha perdido y se pasará al estado WT. El procedimiento que se sigue para pasar de WT a CT es análogo. La transiciones L_t (*Loss*) y R_t (*Recovery*) entre los estados quedan entonces definidas como sigue:

$$L_t = \begin{cases} 0 & \text{si } change_t = 0 \vee (change_t = 1 \wedge f_t^{(appearance)} < \tau) \\ 1 & \text{si } change_t = 1 \wedge f_t^{(appearance)} > \tau \end{cases} \quad (3.3)$$

$$R_t = \begin{cases} 0 & \text{si } change_t = 0 \vee (change_t = 1 \wedge f_t^{(appearance)} > \tau) \\ 1 & \text{si } change_t = 1 \wedge f_t^{(appearance)} < \tau \end{cases} \quad (3.4)$$

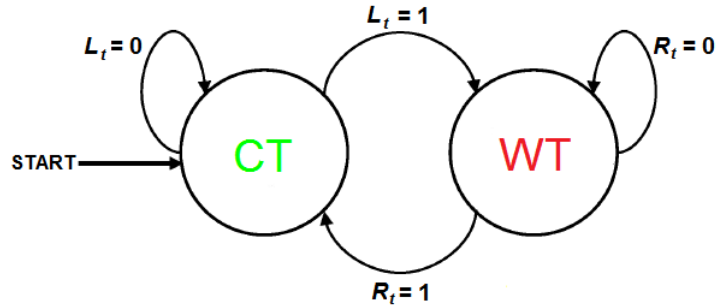


Figura 3.2: Máquina de estados para la decisión CT/WT en cada *frame*.

⁴Esta máquina de estados sigue la misma idea que la utilizada en [8] para realizar la segmentación temporal.

⁵Se sigue la misma idea que en [42].

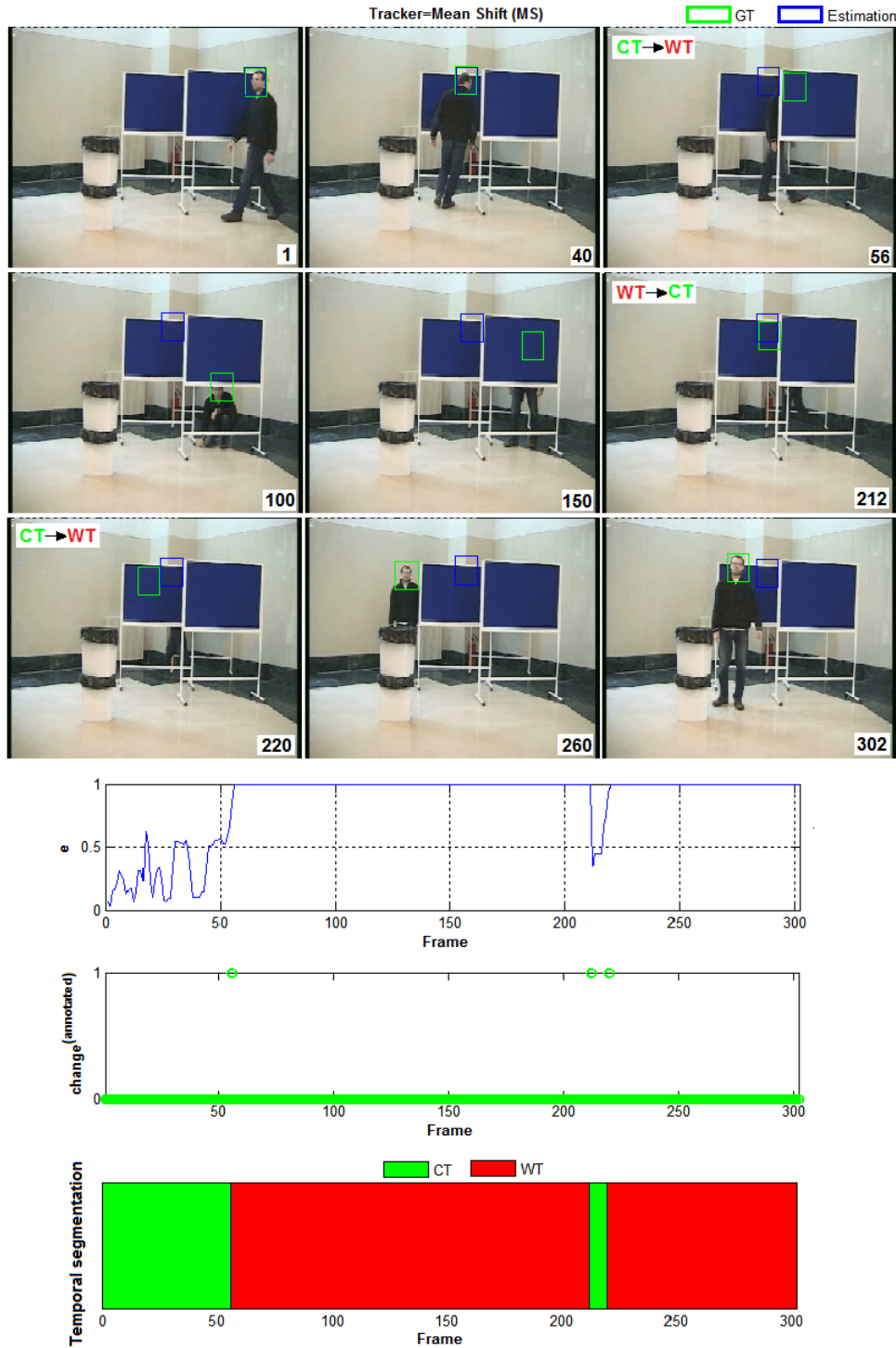


Figura 3.3: Ejemplo de secuencia con *frames* de cambio CT→WT y WT→CT. Arriba se muestran los *frames* 1, 40, 56, 100, 150, 212, 220, 260 y 302. En las gráficas de abajo se representa la señal de error de GT, la señal de cambio anotada y la segmentación temporal CT/WT. El objeto se pierde en el *frame* 56 (cambio CT→WT), se recupera en el 212 (cambio WT→CT) y se vuelve a perder en el 220 (cambio CT→WT). (*Tracker*: *Mean Shift* (MS) [43])

3.2. Detección de anomalías

Desde un punto de vista genérico, el término de detección de anomalías [44] se refiere al problema de encontrar patrones que no están conformes con un comportamiento esperado o normal. En este trabajo se emplean modelos de seguimiento correcto (CT) obtenidos a partir de las características y se consideran anomalías a los valores poco probables de dichos modelos. Estos valores serán los que indiquen la presencia de un posible *frame* de cambio. Para llevar a cabo la detección de las anomalías se plantean tres variantes de una misma técnica que consta de dos fases (ver Figura 3.4):

- **Fase de entrenamiento.** Esta fase se divide en dos partes:
 - **Generación del modelo de CT.** Utilizando un conjunto de secuencias de entrenamiento y las anotaciones de GT asociadas se genera un modelo de CT. Como resultado se obtienen los parámetros relativos a la función de densidad de probabilidad (f.d.p.) del modelo.
 - **Obtención del umbral óptimo.** Para esta parte se selecciona un grupo distinto de secuencias de validación cruzada que contengan *frames* de cambio. El objetivo es encontrar, para la f.d.p obtenida, el umbral de probabilidad óptimo ε_{opt} que permita discriminar mejor los valores anómalos (poco probables) asociados a los *frames* de cambio. Para ello se realiza un barrido de posibles umbrales de probabilidad ε en la f.d.p del modelo y se hace uso de los datos de GT para anotar los *frames* de cambio.
- **Fase de test.** En las secuencias de test se detectan los *frames* de cambio utilizando el umbral ε_{opt} obtenido en la fase anterior.

En el siguiente apartado se detallan las tres posibles versiones de la técnica.

3.2.1. Versiones propuestas

1. **Un modelo de CT por cada característica.** En esta versión se utilizan individualmente los modelos de CT de cada característica, los cuales son especificados por su f.d.p. $p(f)$. El umbral $\varepsilon_{opt(1)}$ será aquél que proporcione el valor más alto de *F-score* (F)⁶. Es decir, $\varepsilon_{opt(1)}$ habrá de cumplir:

$$F_{\varepsilon_{opt(1)}} = \max \{F_{\varepsilon} : \varepsilon > 0\} \quad (3.5)$$

En las secuencias de test se detecta un cambio en los *frames* t en los que el valor de la característica tenga una probabilidad menor que el umbral $\varepsilon_{opt(1)}$. De esta manera la señal

⁶Las métricas *Precision* (P), *Recall* (R) y *F-score* (F) se describen en la Sección 5.4

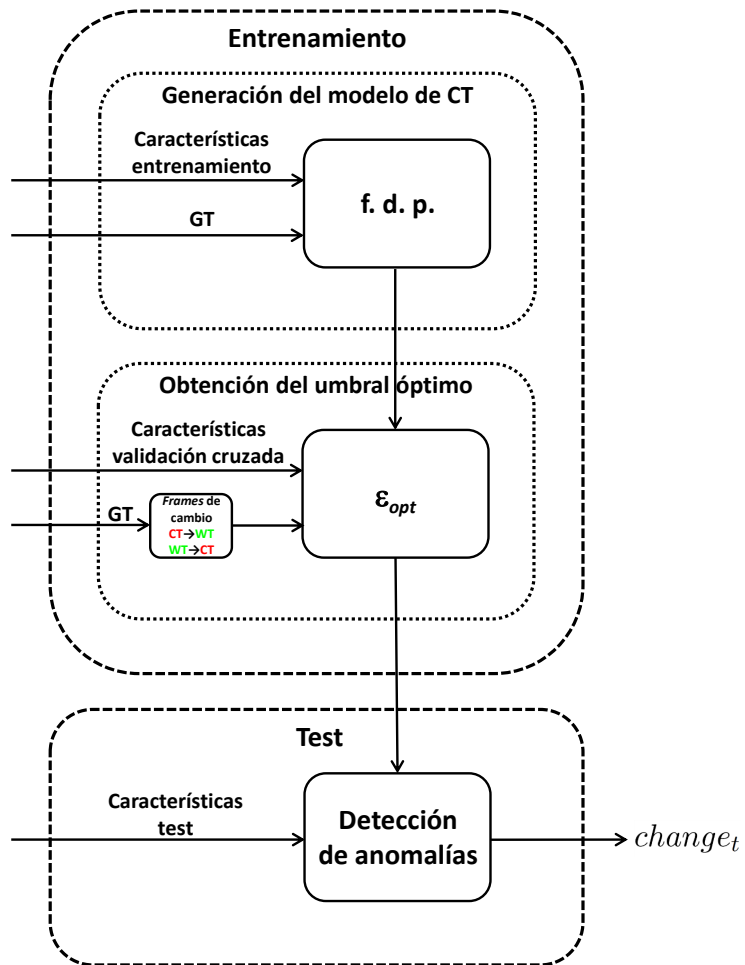


Figura 3.4: Diagrama de bloques de la técnica de detección de anomalías.

de cambio se define como sigue:

$$change_t = \begin{cases} 1 & \text{si } p(f_t) < \varepsilon_{opt(1)} \\ 0 & \text{resto} \end{cases} \quad (3.6)$$

2. **Un modelo de CT para todas las características.** En esta versión se emplea un único modelo de CT multivariante creado a partir de los modelos de CT de cada característica. Asumiendo independencia entre las mismas, la f.d.p. del modelo multivariante se calcula como producto de las f.d.p. de los modelos de cada característica:

$$\tilde{p}(f^{(1)}, \dots, f^{(n)}) = \prod_{i=1}^n p(f^{(i)}) \quad (3.7)$$

Al igual que en la versión anterior, el umbral $\varepsilon_{opt(2)}$ será aquél con el que se consiga el mayor F. Nuevamente $\varepsilon_{opt(2)}$ habrá de cumplir:

$$F_{\varepsilon_{opt(2)}} = \max \{F_{\varepsilon} : \varepsilon > 0\} \quad (3.8)$$

En las secuencias de test se detecta un cambio en los *frames* t cuyo vector de características $\mathbf{f}_t = (f_t^{(1)}, \dots, f_t^{(n)})$ tenga una probabilidad menor que el umbral $\varepsilon_{opt(2)}$. En este caso la señal de cambio será:

$$change_t = \begin{cases} 1 & \text{si } \tilde{p}(\mathbf{f}_t) < \varepsilon_{opt(2)} \\ 0 & \text{resto} \end{cases} \quad (3.9)$$

3. **Combinación de las detecciones de cada característica.** En esta versión el objetivo es combinar las detecciones realizadas individualmente con los modelos de CT de cada característica. Para la f.d.p. $p(f)$ de cada modelo se elige un umbral de probabilidad $\varepsilon_{opt(3)}$ con el que se consiga una elevada *Precision* (P)⁶. Escogiendo este umbral se intenta evitar que haya muchas falsas detecciones. Puesto que se van a sumar las detecciones de todas las características no es deseable que haya demasiadas erróneas ya que empeorarían los resultados finales. Para obtener dicho umbral se seleccionan, en primer lugar, aquellos umbrales que proporcionen una P mayor o igual que una determinada precisión mínima (P_{min}) que deberá ser alta. Posteriormente se elige de todos ellos aquél con el que se obtenga el máximo *Recall* (R)⁶. Es decir, $\varepsilon_{opt(3)}$ habrá de verificar:

$$R_{\varepsilon_{opt(3)}} = \max \{R_{\varepsilon} : \varepsilon \in \{\varepsilon > 0 : P \geq P_{min}\}\} \quad (3.10)$$

En caso de existir varios, se optará por el de mayor P. En las secuencias de test se detectan primero los *frames* de cambio utilizando por separado la f.d.p. $p(f^{(i)})$ de cada uno de los

modelos de CT y su respectivo umbral $\varepsilon_{opt(3)}^{(i)}$. Se obtiene así la señal de cambio $change_t^{(i)}$ de la misma manera que en la primera versión de la técnica. Después se suman de manera lógica todas las detecciones realizadas con cada modelo, siendo finalmente la señal de cambio:

$$change_t = \bigvee_{i=1}^n change_t^{(i)} = change_t^{(1)} \vee \dots \vee change_t^{(n)} \quad (3.11)$$

Mediante esta combinación se busca mantener la elevada P y obtener un R más alto que el conseguido individualmente con cada característica.

Capítulo 4

Características

En este capítulo se presentan las características seleccionadas para detectar los *frames* de cambio. Todas ellas se usan ampliamente en el ámbito de procesamiento de imágenes. Las dos primeras sirven para analizar los cambios de forma del objeto (Sección 4.1). Las dos siguientes se emplean para estudiar su movimiento (Sección 4.2). El resto se utilizan para medir la estabilidad de determinadas propiedades relativas a la apariencia (Sección 4.3). Tras explicar con detalle todas las características se aporta un resumen de los aspectos más importantes de cada una (Sección 4.4).

4.1. Forma

En primer lugar se describen las características referentes a la forma del objeto (ver Figura 4.1): variación de la relación de aspecto (Sección 4.1.1) y del área (Sección 4.1.2).

4.1.1. Variación de la relación de aspecto

El objetivo de esta característica¹ es detectar cambios repentinos en la forma, los cuales podrían indicar una pérdida o recuperación del objeto. Siendo la relación de aspecto $R = \frac{W}{H}$ (donde W y H son, respectivamente, el ancho y el alto del *bounding box* que contiene al objeto), el valor de la característica entre dos *frames* consecutivos, $t - 1$ y t , se calcula de la siguiente manera:

$$shape\ ratio\ score = 1 - \frac{\min\{R_t, R_{t-1}\}}{\max\{R_t, R_{t-1}\}} \quad (4.1)$$

¹Estas características son adaptaciones de las propuestas en [11]. En este trabajo se ha tomado como medida el valor complementario de dichas características. En el caso de *histogram difference* además se ha realizado una normalización previa. Con estas modificaciones el valor 0 indicará igualdad y a medida que los valores vayan siendo más cercanos a 1 significará que existe más diferencia. De esta manera se sigue la misma convención para todas las características, lo que aportará más sencillez al estudio realizado.

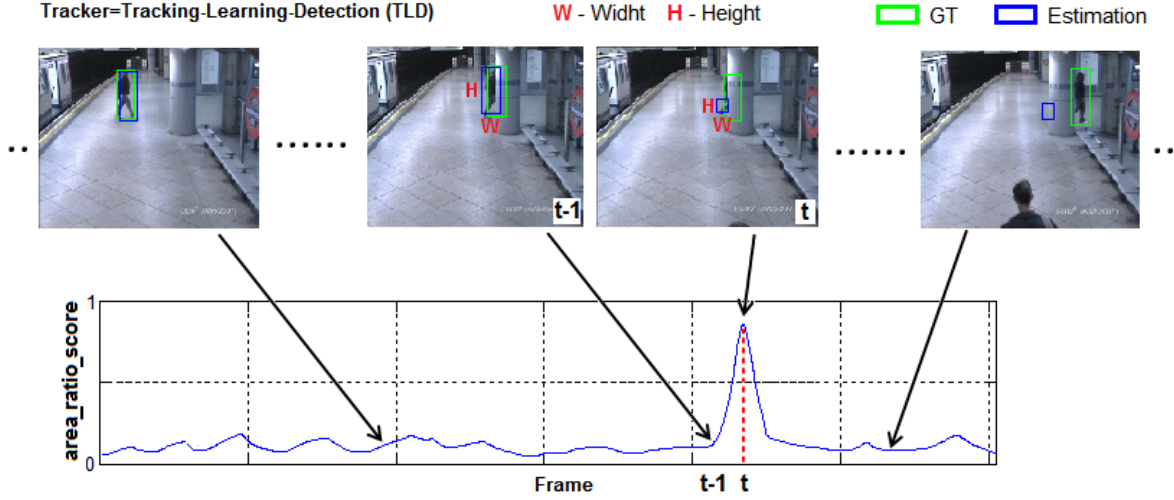


Figura 4.1: Ejemplo de uso de las características relacionadas con la forma. El propósito es identificar cambios repentinos que podrían indicar una pérdida o recuperación del objeto. (*Tracker: Tracking-Learning-Detection* (TLD) [45])

Si la forma del objeto seguido se mantiene constante, la característica tenderá a tomar valores cercanos a 0. En caso de producirse algún cambio repentino, el valor tenderá a 1.

4.1.2. Variación del área

Al igual que la medida anterior, la finalidad de esta característica¹ es encontrar cambios bruscos en el área del *bounding box* que contiene al objeto. Siendo ésta $A = W \cdot H$, el valor de la característica entre dos *frames* consecutivos, $t - 1$ y t , se obtiene empleando la siguiente fórmula:

$$area\ ratio\ score = 1 - \frac{\min \{A_t, A_{t-1}\}}{\max \{A_t, A_{t-1}\}} \quad (4.2)$$

En el caso de que la forma del objeto no presente muchas variaciones, los valores de la característica serán próximos a 0. Si tuviera lugar algún cambio brusco, el valor tenderá a 1.

4.2. Movimiento

A continuación se especifican las características relativas al movimiento del objeto (ver Figura 4.2): suavidad en la dirección (Sección 4.2.1) y en el desplazamiento (Sección 4.2.2).

4.2.1. Suavidad en la dirección

Considerando que la trayectoria del objeto que se sigue es regular, esta característica (puesta en [11]) se empleará para detectar cambios repentinos en la dirección estimada por el

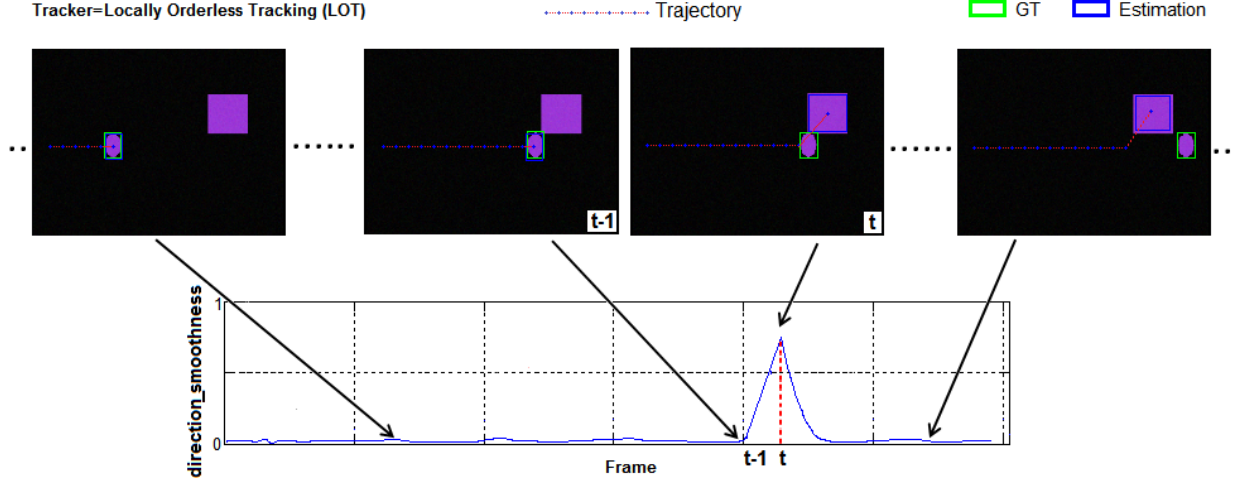


Figura 4.2: Ejemplo de uso de las características relacionadas con el movimiento. El objetivo es examinar la dirección y el desplazamiento para detectar cambios bruscos que podrían ser debidos a una pérdida o recuperación del objeto. (*Tracker: Locally Orderless Tracking (LOT)* [46])

tracker. Estos cambios podrían ser debidos a una pérdida o recuperación del objeto. Si se llama $\Delta\theta_t$ a la diferencia entre los ángulos que forman con el eje x las direcciones en los *frames* $t-1$ y t (siendo $0^\circ \leq \Delta\theta_t \leq 180^\circ$), el valor de la característica se calcula del siguiente modo:

$$direction\ smoothness = \frac{\Delta\theta_t}{180^\circ} \quad (4.3)$$

Cuando la trayectoria estimada sea estable, la característica adquirirá valores próximos a 0. Si en algún momento la dirección cambia repentinamente, el valor aumentará, llegando a ser 1 si el sentido es el opuesto.

4.2.2. Suavidad en el desplazamiento

Mediante el uso de esta característica¹ se pretende comprobar si el último desplazamiento del objeto seguido se aproxima al valor medio de todos los desplazamientos anteriores. Siendo \mathbf{P}_{t-1} y \mathbf{P}_t , respectivamente, las posiciones del objeto en los *frames* $t-1$ y t , el último desplazamiento será $s_t = \|\mathbf{P}_t - \mathbf{P}_{t-1}\|$. Comparando este valor con el desplazamiento promedio $\bar{s} = \frac{1}{t-1} \sum_{i=1}^{t-1} s_i$ se obtiene el valor de la característica:

$$speed\ smoothness = 1 - \frac{\min\{s_t, \bar{s}\}}{\max\{s_t, \bar{s}\}} \quad (4.4)$$

Si los desplazamientos realizados tienden a ser constantes, los valores que tomará la característica se aproximarán a 0. En caso de producirse algún desplazamiento que se desvíe mucho de la media, el valor se acercará a 1. Cuando se detecte una parada, el valor será igual a 1.

4.3. Apariencia

En los siguientes apartados se definen las características relacionadas con la apariencia del objeto (ver Figura 4.3): diferencia entre los histogramas (Sección 4.3.1), SSD (Sección 4.3.2), diferencia de textura (Sección 4.3.3), SSIM (Sección 4.3.4), diferencia entre los HOG (Sección 4.3.5), diferencia entre los CLD (Sección 4.3.6) y diferencia entre las matrices de covarianzas (Sección 4.3.7).

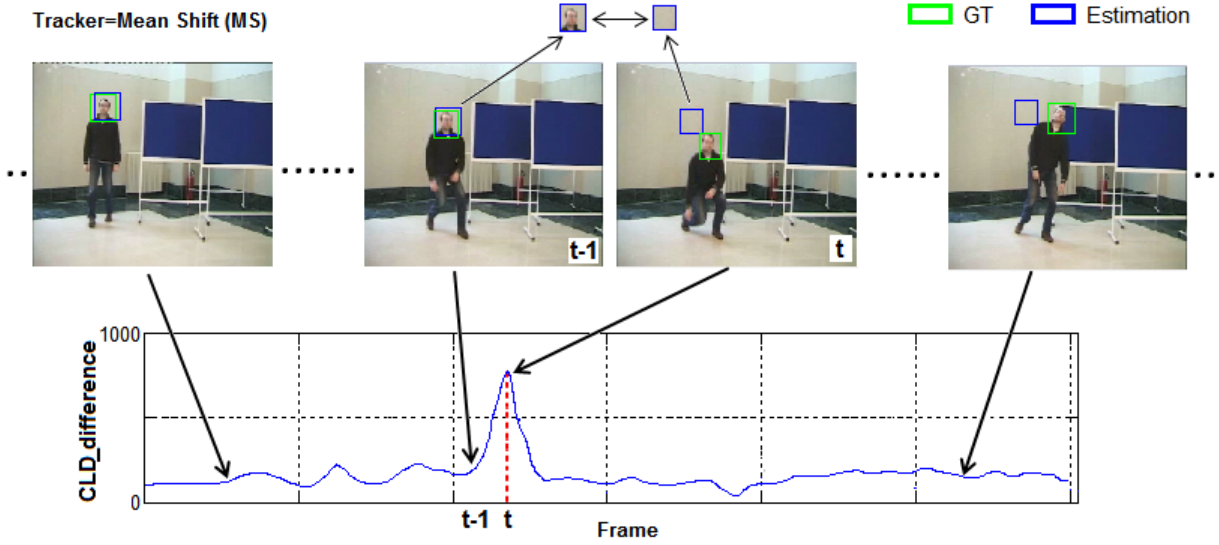


Figura 4.3: Ejemplo de uso de las características relacionadas con la apariencia. El objetivo es medir la estabilidad de determinadas propiedades para detectar variaciones que podrían indicar una pérdida o recuperación del objeto. (*Tracker: Mean Shift* (MS) [43])

4.3.1. Diferencia entre los histogramas

Esta característica¹ servirá para medir la diferencia entre las imágenes del objeto estimadas en *frames* consecutivos. Para realizar esa comparación se utilizarán los histogramas \mathbf{H}_{t-1} y \mathbf{H}_t , correspondientes a las respectivas imágenes. Se analizarán independientemente los tres canales RGB y la versión en escala de grises usando la siguiente fórmula:

$$histogram\ difference = 1 - \frac{1}{256} \sum_{i=1}^{256} \frac{\min \{ \mathbf{H}_t(i), \mathbf{H}_{t-1}(i) \}}{\max \{ \mathbf{H}_t(i), \mathbf{H}_{t-1}(i) \}} \quad (4.5)$$

Si las dos imágenes se parecen, el valor de la característica será cercano a 0. Cuanto mayor sea la diferencia entre ambas, más se aproximará el valor a 1.

4.3.2. SSD (*Sum of Squared Differences*)

Esta característica mide la similitud entre las imágenes del objeto estimadas en *frames* consecutivos mediante la suma de diferencias al cuadrado [47]. Siendo \mathbf{I}_{t-1} e \mathbf{I}_t , respectivamente, las imágenes del objeto estimadas en los *frames* $t - 1$ y t , el valor de la característica se calcula como:

$$SSD = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (\mathbf{I}_t(x, y) - \mathbf{I}_{t-1}(x, y))^2, \quad (4.6)$$

donde x e y se refieren a las coordenadas de los píxeles y M y N son las dimensiones de las imágenes.

Cuando las dos imágenes se parezcan el valor de la característica se acercará a 0. Según vaya aumentando la diferencia, el valor será más alto, no existiendo una cota superior.

4.3.3. Diferencia de textura

La finalidad de esta característica (propuesta en [11]) es comparar las imágenes del objeto estimadas en *frames* consecutivos, midiendo para ello la diferencia entre sus respectivas texturas. Para extraer la textura de cada una de las imágenes se empleará un banco de filtros de Gabor [48] con diferentes escalas (2, 4, 8 y 16) y orientaciones (0° , 45° , 90° y 135°) (ver Figura 4.4). Los filtros de Gabor son una familia de filtros lineales paso-banda cuya respuesta al impulso es una función gaussiana modulada por una sinusoidal:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cdot \exp\left(i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right), \quad i = \sqrt{-1}, \quad (4.7)$$

$$\begin{aligned} x' &= x \cos(\theta) + y \sin(\theta) \\ y' &= -x \sin(\theta) + y \cos(\theta) \end{aligned} \quad (4.8)$$

donde x e y son las coordenadas en dominio espacial, λ representa la longitud de onda de la senoide, θ indica la orientación, ψ es el desfase, σ denota la escala y γ es la relación de aspecto espacial que especifica la elipticidad del filtro.

En ocasiones también se suelen tratar las componentes real e imaginaria del filtro por separado:

$$\Re\{g(x, y; \lambda, \theta, \psi, \sigma, \gamma)\} = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cdot \cos\left(2\pi\frac{x'}{\lambda} + \psi\right) \quad (4.9)$$

$$\Im\{g(x, y; \lambda, \theta, \psi, \sigma, \gamma)\} = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cdot \sin\left(2\pi\frac{x'}{\lambda} + \psi\right) \quad (4.10)$$

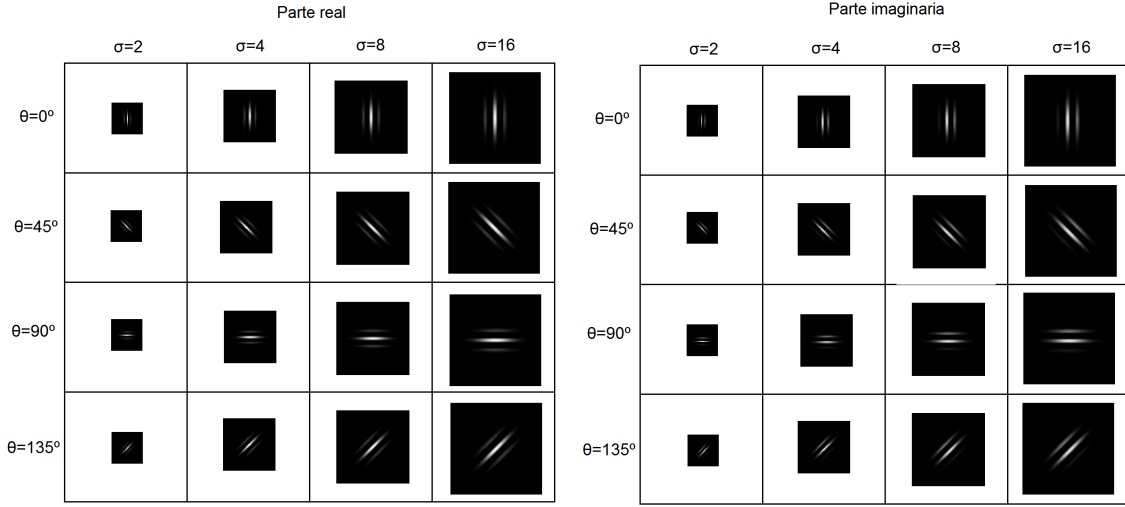


Figura 4.4: Banco de filtros de Gabor con diferentes escalas (2, 4, 8 y 16) y orientaciones (0° , 45° , 90° y 135°). A la izquierda se representa la parte real de los filtros y a la derecha la parte imaginaria.

Aplicando los 16 (4×4) filtros del banco a ambas imágenes, se genera para cada una el vector de características de textura \mathbf{v} . Este vector, de 32 elementos, estará formado por la media μ_{ij} y la varianza σ_{ij}^2 de la imagen filtrada con cada uno de los filtros:

$$\mathbf{v} = \begin{pmatrix} \mu_{11} \\ \sigma_{11}^2 \\ \mu_{12} \\ \sigma_{12}^2 \\ \vdots \\ \mu_{44} \\ \sigma_{44}^2 \end{pmatrix} \quad (4.11)$$

Si se llaman \mathbf{v}_{t-1} y \mathbf{v}_t , respectivamente, a los vectores de características de textura relativos a las imágenes del objeto estimadas en los *frames* $t-1$ y t , el valor de la característica se calcula como la norma euclídea del vector diferencia:

$$texture\ difference = \sqrt{\sum_{i=1}^{32} (\mathbf{v}_t(i) - \mathbf{v}_{t-1}(i))^2} \quad (4.12)$$

Cuando el parecido entre las imágenes sea alto, el valor de la característica se aproximará a 0. A medida que haya menos similitud el valor será más alto, no existiendo una cota superior.

4.3.4. SSIM (*Structural SIMilarity*)

El propósito de esta característica es evaluar la similitud entre las imágenes del objeto estimadas en *frames* consecutivos, utilizando la técnica SSIM propuesta en [49]. Se trata de una técnica que mide la calidad que se percibe en una imagen con respecto a otra, la cual se considera que está libre de distorsión. SSIM se fundamenta en la idea de que el sistema visual humano está altamente capacitado para extraer información estructural de una imagen y considera que se puede obtener una buena aproximación de la calidad percibida a través de una medida de su degradación.

Para calcular la similitud entre las dos imágenes SSIM tiene en cuenta tres aspectos: luminancia, contraste y estructura. La comparativa se realiza localmente entre las distintas regiones de las imágenes. Considerando que \mathbf{a} y \mathbf{b} son dos ventanas locales pertenecientes, respectivamente, a las imágenes del objeto estimadas en los *frames* $t - 1$ y t , se obtiene cada valor del mapa SSIM como:

$$SSIM(\mathbf{a}, \mathbf{b}) = l(\mathbf{a}, \mathbf{b}) \cdot c(\mathbf{a}, \mathbf{b}) \cdot s(\mathbf{a}, \mathbf{b}), \quad (4.13)$$

donde $l(\mathbf{a}, \mathbf{b})$, $c(\mathbf{a}, \mathbf{b})$ y $s(\mathbf{a}, \mathbf{b})$ son, respectivamente, las medidas comparativas de luminancia, contraste y estructura (ver Figura 4.5).

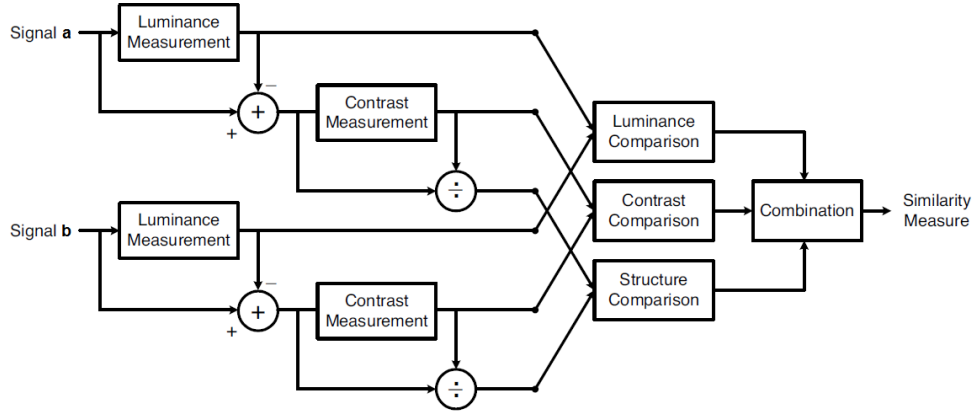


Figura 4.5: Diagrama de bloques del método SSIM (extraído de [49]).

La función que se utiliza para comparar la luminancia se define como:

$$l(\mathbf{a}, \mathbf{b}) = \frac{2\mu_a\mu_b + C_1}{\mu_a^2 + \mu_b^2 + C_1} = \frac{2(1+R)}{1 + (1+R)^2 + \frac{C_1}{\mu_a^2}}, \quad (4.14)$$

$$\mu_a = \frac{1}{N} \sum_{i=1}^N a_i, \quad \mu_b = \frac{1}{N} \sum_{i=1}^N b_i = (1+R)\mu_a, \quad (4.15)$$

donde a_i y b_i denotan, respectivamente, la intensidad de cada uno de los píxeles pertenecientes a las ventanas locales \mathbf{a} y \mathbf{b} , N indica el número de píxeles de la ventana, μ_a y μ_b son los valores medios de intensidad de los píxeles, R representa el cambio de luminancia relativo a μ_a y C_1 es una constante cuyo valor es prácticamente despreciable. Observando las fórmulas anteriores se comprueba que si $\mu_a \simeq \mu_b$ entonces R será muy pequeño y, por tanto, el valor de $l(\mathbf{a}, \mathbf{b})$ será alto, lo que indicará que existe una elevada similitud en luminancia.

Para comparar el contraste se emplea la siguiente función:

$$c(\mathbf{a}, \mathbf{b}) = \frac{2\sigma_a\sigma_b + C_2}{\sigma_a^2 + \sigma_b^2 + C_2}, \quad (4.16)$$

$$\sigma_a = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (a_i - \mu_a)^2}, \quad \sigma_b = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (b_i - \mu_b)^2}, \quad (4.17)$$

donde σ_a y σ_b son las desviaciones típicas de intensidad de los píxeles y C_2 es una constante de valor casi despreciable. En esta ocasión se aprecia que si $\sigma_a \simeq \sigma_b$ entonces el valor de $c(\mathbf{a}, \mathbf{b})$ será alto, lo que significará que hay una elevada similitud en contraste.

En el caso de la estructura la función que se usa para la comparativa es la siguiente:

$$s(\mathbf{a}, \mathbf{b}) = \frac{\sigma_{ab} + C_3}{\sigma_a\sigma_b + C_3}, \quad (4.18)$$

$$\sigma_{ab} = \frac{1}{N-1} \sum_{i=1}^N (a_i - \mu_a)(b_i - \mu_b), \quad (4.19)$$

donde σ_{ab} es la covarianza entre los valores de intensidad de los píxeles y $C_3 = \frac{C_2}{2}$ es una constante de valor prácticamente despreciable. Se puede comprobar que si σ_{ab} es alta entonces el valor de $s(\mathbf{a}, \mathbf{b})$ también lo será, lo que querrá decir que la similitud en estructura es elevada.

Puesto que en la práctica lo que interesa es tener un único valor que determine la similitud entre las dos imágenes, lo que se hace es obtener la media de los valores del mapa SSIM:

$$MSSIM = \frac{1}{M} \sum_{m=1}^M SSIM(\mathbf{a}_m, \mathbf{b}_m), \quad (4.20)$$

donde \mathbf{a}_m y \mathbf{b}_m denotan la m -ésima ventana local de cada imagen y M es el número de valores del mapa (que es igual al número de ventanas locales). El rango de valores de esta medida es $[-1, 1]$, alcanzándose el valor 1 en el caso de que las imágenes sean idénticas (ver ejemplos Figura 4.6). Para seguir el mismo criterio que en el resto de características (en las que el valor 0 indica similitud total), se utilizará finalmente la métrica *DSSIM* (*Structural DisSIMilarity*) definida en [50], que no es más que una transformación lineal de *MSSIM* al intervalo $[0, 1]$:

$$DSSIM = \frac{1 - MSSIM}{2} \quad (4.21)$$

De esta manera el valor de la característica se acercará a 0 si la similitud entre las dos imágenes es alta y, según vaya existiendo más diferencia, el valor se aproximará a 1.

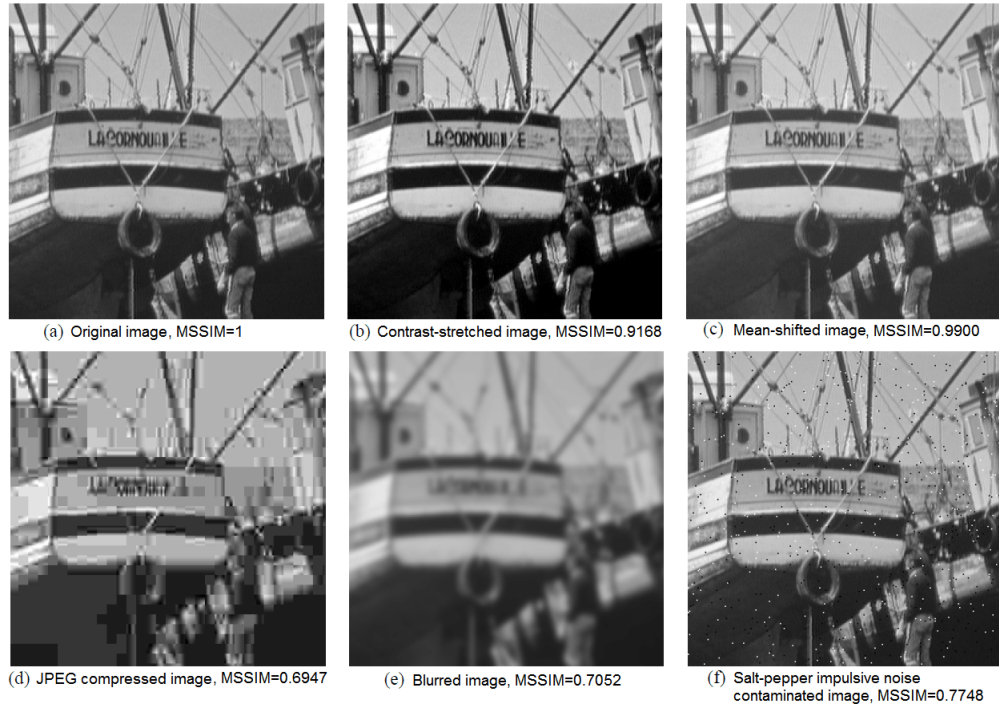


Figura 4.6: Ejemplo de comparación de imágenes con distintos tipos de distorsión utilizando MSSIM (extraído de [49]).

4.3.5. Diferencia entre los HOG (*Histograms of Oriented Gradients*)

Esta característica se empleará para comparar las imágenes del objeto estimadas en *frames* consecutivos, para lo cual se medirá la diferencia entre sus descriptores HOG [51]. El descriptor HOG se basa en que la forma de un objeto en una imagen puede ser representada por la distribución de la orientación de los gradientes.

El primer paso para obtener el descriptor HOG consiste en calcular las derivadas parciales de la imagen en las direcciones x e y . De esta manera se obtiene la orientación y la magnitud del gradiente en cada píxel. Los gradientes proporcionarán información acerca de los contornos, es decir, indicarán el cambio de intensidad entre píxeles contiguos.

Posteriormente se divide la imagen en bloques descriptores y éstos, a su vez, se subdividen en celdas. Para cada celda se calcula un histograma de gradientes orientados. Los valores resultantes se normalizan para mejorar la invarianza frente a cambios de iluminación o sombras.

Los histogramas de cada celda se combinan formando un descriptor del bloque. Por último se concatenan los descriptores de cada bloque en un único descriptor de la imagen. En la Figura 4.7 se muestra de manera esquemática el proceso seguido.

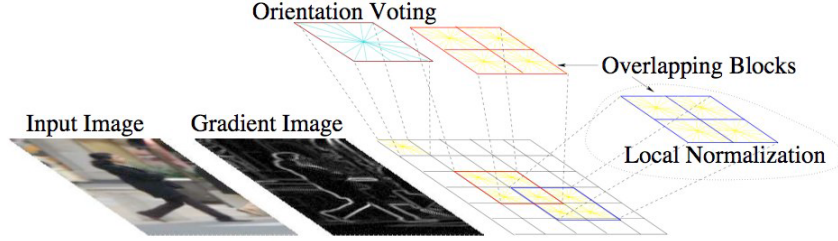


Figura 4.7: Proceso de cálculo de los descriptores HOG (extraído de [52]).

Siendo $\hat{\mathbf{H}}_{t-1}$ y $\hat{\mathbf{H}}_t$, respectivamente, los HOG normalizados de las imágenes del objeto estimadas en los *frames* $t-1$ y t , el valor de la característica se obtiene como la distancia de Bhattacharya [53] entre ambos histogramas:

$$HOG\ difference = \sqrt{1 - \rho [\hat{\mathbf{H}}_t, \hat{\mathbf{H}}_{t-1}]}, \quad (4.22)$$

$$\rho [\hat{\mathbf{H}}_t, \hat{\mathbf{H}}_{t-1}] = \sum_{i=1}^{N_{bins}} \sqrt{\hat{\mathbf{H}}_t(i) \cdot \hat{\mathbf{H}}_{t-1}(i)}, \quad (4.23)$$

donde N_{bins} hace referencia al número de bins de los histogramas.

Si las dos imágenes se parecen, el valor de la característica se aproximará a 0. A medida que la diferencia sea mayor, el valor se tenderá a 1.

4.3.6. Diferencia entre los CLD (*Color Layout Descriptors*)

El objetivo de esta característica es medir la diferencia entre las imágenes del objeto estimadas en *frames* consecutivos comparando sus respectivos descriptores CLD [54]. El descriptor CLD, que se encuentra recogido en el estándar MPEG-7 [55], fue diseñado para capturar la distribución espacial del color en una imagen. Se caracteriza por ser una representación del color muy compacta que presenta una resolución invariante respecto del tamaño de la imagen.

El proceso que se sigue para obtener el CLD de cada una de las imágenes consta de cuatro etapas (ver Figura 4.8):

- **División de la imagen.** En la primera etapa se divide la imagen original en 64 bloques (8 x 8). Considerando que las dimensiones de la imagen son $M \times N$, el tamaño de cada bloque será entonces $\frac{M}{8} \times \frac{N}{8}$. El objetivo de esta fase es garantizar la invariabilidad de la resolución.

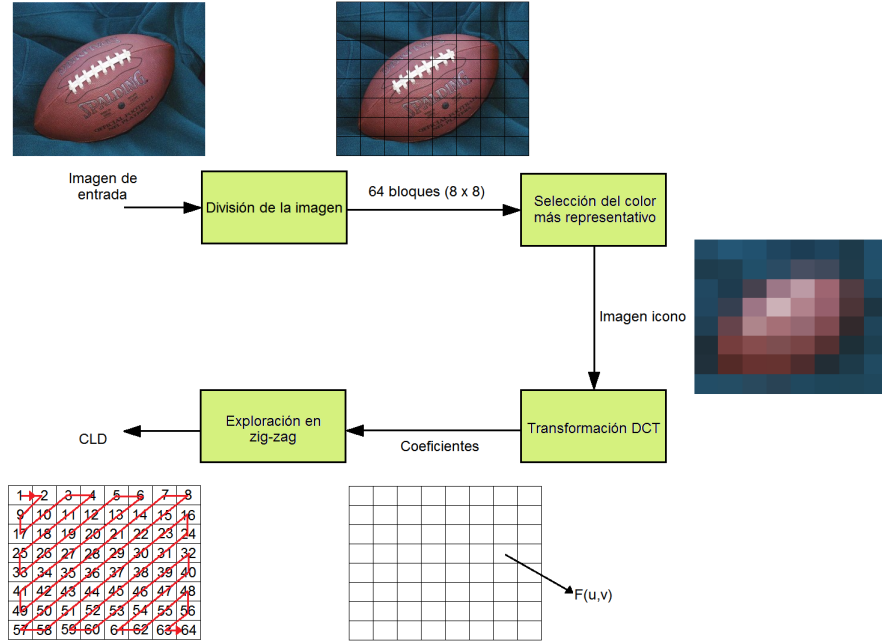


Figura 4.8: Proceso de extracción del CLD.

- **Selección del color más representativo.** El siguiente paso consiste en escoger un color representativo para cada uno de los 64 bloques. Aunque existen varios métodos para realizar esta selección, se recomienda el uso de la media de la intensidad de los píxeles de cada bloque como color más representativo, ya que es un método sencillo con una precisión que, en general, es suficiente. A la imagen resultante (imagen icono) de 64 píxeles se le aplica la conversión del espacio original de color RGB al espacio YCbCr (donde Y es la componente de luminancia y Cb y Cr son, respectivamente, las componentes de cromaticidad de diferencia de azul y diferencia de rojo).
- **Transformación DCT.** En la tercera fase se aplica la transformada DCT (*Discrete Cosine Transform*) a cada una de las componentes YCbCr. De esta manera se obtienen tres matrices de 8 x 8 coeficientes DCT. Las fórmulas que se emplean para calcular la DCT de la matriz de cada componente son las siguientes:

$$F(u, v) = \alpha_u \alpha_v \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} A_{ij} \cos\left(\frac{\pi(2i+1)u}{2M}\right) \cdot \cos\left(\frac{\pi(2j+1)v}{2N}\right), \quad \begin{matrix} 0 \leq u \leq M-1 \\ 0 \leq v \leq N-1 \end{matrix} \quad (4.24)$$

$$\alpha_u = \begin{cases} \frac{1}{\sqrt{M}} & u = 0 \\ \sqrt{\frac{2}{M}} & 1 \leq u \leq M-1 \end{cases}, \quad \alpha_v = \begin{cases} \frac{1}{\sqrt{N}} & v = 0 \\ \sqrt{\frac{2}{N}} & 1 \leq v \leq N-1 \end{cases}, \quad (4.25)$$

donde los índices (i, j) y (u, v) hacen referencia, respectivamente, a las coordenadas en el

dominio espacial y frecuencial, $M = 8$ y $N = 8$ son las dimensiones de la matriz y A_{ij} denota la intensidad de cada píxel.

- **Exploración en zig-zag.** Por último se realiza un recorrido en zig-zag por los coeficientes DCT de las tres matrices, tal y como se muestra en la Figura 4.8. El motivo por el cual sigue este orden en la exploración es porque en la parte superior izquierda de las matrices están ubicados los coeficientes de baja frecuencia que concentran la mayor parte de la energía de la imagen. Como resultado se obtienen tres vectores (\mathbf{DY} , \mathbf{DCb} y \mathbf{DCr}) cuyas primeras componentes albergarán casi toda la información. Estos tres vectores son los que formarán el descriptor CLD.

Si se denominan $\mathbf{CLD}^{t-1} = \{\mathbf{DY}^{t-1}, \mathbf{DCb}^{t-1}, \mathbf{DCr}^{t-1}\}$ y $\mathbf{CLD}^t = \{\mathbf{DY}^t, \mathbf{DCb}^t, \mathbf{DCr}^t\}$, respectivamente, a los descriptores de las imágenes del objeto estimadas en los *frames* $t - 1$ y t , la fórmula que se utilizará para llevar a cabo la comparación es la siguiente:

$$CLD\ difference = \sqrt{\sum_{i=1}^{64} \omega_i^Y (DY_i^t - DY_i^{t-1})^2} + \sqrt{\sum_{j=1}^{64} \omega_j^{Cb} (DCb_j^t - DCb_j^{t-1})^2} + \sqrt{\sum_{k=1}^{64} \omega_k^{Cr} (DCr_k^t - DCr_k^{t-1})^2}, \quad (4.26)$$

donde i , j y k son los índices de los coeficientes de los tres vectores que forman los descriptores y ω_i^Y , ω_j^{Cb} y ω_k^{Cr} son los pesos que se utilizan para ponderar cada uno de los coeficientes. Estos pesos deberán ser escogidos adecuadamente para dar más importancia a los coeficientes de baja frecuencia. En la Tabla 4.1 se muestran los pesos usados en este trabajo.

El valor de la característica se aproximará a 0 si el parecido entre las dos imágenes es alto. Cuanto mayor sea la diferencia entre ambas, más alto será su valor, no existiendo una cota superior.

	ω_1	ω_2	ω_3	ω_4	\dots	ω_{64}
Y	2	2	2	1	1	1
Cb	2	1	1	1	1	1
Cr	4	2	2	1	1	1

Tabla 4.1: Pesos utilizados para los coeficientes del descriptor CLD.

4.3.7. Diferencia entre las matrices de covarianzas

El propósito de esta característica es medir la diferencia entre las imágenes del objeto estimadas en *frames* consecutivos, empleando para ello las matrices de covarianzas [56] asociadas a dichas imágenes. Las ventajas de utilizar la matriz de covarianzas como descriptor de una imagen

son varias: permite fusionar múltiples características sin que sea necesaria ninguna normalización; además las dimensiones de la matriz siempre son las mismas, independientemente de cuál sea el tamaño de la imagen; asimismo la matriz de covarianzas es invariante a las rotaciones y cambios de iluminación.

El primer paso para obtener la matriz de covarianzas de cada una de las imágenes es construir, para cada píxel k , el vector de características \mathbf{f}_k . Este vector estará formado por las coordenadas x e y del píxel y los valores de las tres intensidades RGB²:

$$\mathbf{f}_k = \begin{pmatrix} x \\ y \\ \mathbf{I}^R(x, y) \\ \mathbf{I}^G(x, y) \\ \mathbf{I}^B(x, y) \end{pmatrix} \quad (4.27)$$

A partir de los vectores de características \mathbf{f}_k de todos los píxeles de la imagen, se obtiene la matriz de covarianzas:

$$\mathbf{C} = \frac{1}{MN} \sum_{k=1}^{MN} (\mathbf{f}_k - \boldsymbol{\mu})(\mathbf{f}_k - \boldsymbol{\mu})^T, \quad (4.28)$$

donde M y N son las dimensiones de la imagen y $\boldsymbol{\mu}$ es el vector de medias de las cinco características. De esta manera se consigue representar una imagen de tamaño $M \times N$ mediante una matriz 5×5 . Esta matriz es simétrica y los elementos de su diagonal principal representan la varianza de cada una de las características. El resto de elementos indican las correlaciones existentes entre cada una de ellas.

Considerando que \mathbf{C}_{t-1} y \mathbf{C}_t son, respectivamente, las matrices de covarianzas de las imágenes del objeto estimadas en los *frames* $t-1$ y t , la diferencia entre ambas matrices se obtiene utilizando la siguiente métrica³ (propuesta en [57]):

$$\text{covariance matrix difference} = \sqrt{\sum_{i=1}^5 \log^2(\lambda_i(\mathbf{C}_t, \mathbf{C}_{t-1}))}, \quad (4.29)$$

donde $\{\lambda_i(\mathbf{C}_t, \mathbf{C}_{t-1})\}$ son los autovalores generalizados de \mathbf{C}_t y \mathbf{C}_{t-1} , calculados a partir de la siguiente fórmula:

$$\lambda_i \mathbf{C}_t \mathbf{v}_i - \mathbf{C}_{t-1} \mathbf{v}_i = 0, \quad i = 1, \dots, 5 \quad (4.30)$$

²Se podrían incluir más características relacionadas con la apariencia del objeto (gradiente, infrarrojo, etc.) como se propone en [56].

³Esta métrica asume que las matrices de covarianzas son semidefinidas positivas. Para que esto sea así es necesario que no exista dependencia lineal entre ninguna de las características.

siendo \mathbf{v}_i los autovectores generalizados.

Si las dos imágenes se parecen, el valor que adquirirá característica será cercano a 0. A medida que la diferencia sea mayor el valor aumentará, no existiendo una cota superior.

4.4. Resumen

Para concluir el capítulo se muestra en la Tabla 4.2 una recopilación, a modo de resumen, de los aspectos más relevantes de cada una de las características seleccionadas.

Característica	Tipo	Propiedad	Rango	Uso en <i>tracking</i>	Uso en estimación de fiabilidad
<i>shape ratio score</i>	Forma	Tamaño	$[0, 1)$	-	[11]
<i>area ratio score</i>		Área	$[0, 1)$	-	[11]
<i>direction smoothness</i>	Movimiento	Dirección	$[0, 1]$	-	[11]
<i>speed smoothness</i>		Desplazamiento	$[0, 1]$	-	[11]
<i>histogram difference</i>	Apariencia	Intensidad	$[0, 1)$	-	[11]
<i>SSD</i>			$[0, \infty)$	[47]	-
<i>texture difference</i>		Textura	$[0, \infty)$	-	[11]
<i>DSSIM</i>		Estructura	$[0, 1]$	[50]	-
<i>HOG difference</i>			$[0, 1)$	[51]	-
<i>CLD difference</i>		Color y estructura	$[0, \infty)$	[54]	-
<i>covariance matrix difference</i>			$[0, \infty)$	[56]	-

Tabla 4.2: Resumen de los aspectos más importantes de las características seleccionadas. Los valores altos indicarán la posible presencia de un *frame* de cambio.

Capítulo 5

Resultados experimentales

En este capítulo se muestran los resultados experimentales obtenidos con el algoritmo propuesto en el Capítulo 3. En primer lugar se comentan algunos aspectos referentes a la implementación de las características y del algoritmo (Sección 5.1), se indican los *trackers* sobre los que se ha aplicado el algoritmo (Sección 5.2), se presenta el *dataset* empleado (Sección 5.3) y se especifican las métricas usadas para la evaluación (Sección 5.4). Después se realiza un análisis de las características utilizadas (Sección 5.5) y se exponen los resultados de la evaluación (Sección 5.6). Por último se hace una comparativa con los resultados que consiguen otras técnicas del estado del arte (Sección 5.7).

5.1. Implementación de las características y del algoritmo

Para la implementación de las características y del algoritmo se ha utilizado el entorno Matlab. Todas las características se han programado íntegramente a excepción de *DSSIM*, *HOG difference* y *SSD*. En estos tres casos se ha hecho uso de material *online*¹²³. El código del algoritmo también ha sido desarrollado al completo.

5.2. *Trackers*

El algoritmo propuesto se ha aplicado sobre seis *trackers*: *Mean Shift* (MS) [43], *Color-based Particle Filter* (CPF) [41], *Locally Orderless Tracking* (LOT) [46], *Incremental Visual Tracking* (IVT) [58], *SuperPixel Tracking* (SPT) [59] y *Tracking-Learning-Detection* (TLD) [45]. Todos ellos están implementados en Matlab. El código de los dos primeros ha sido proporcionado por

¹<https://ece.uwaterloo.ca/~z70wang/research/ssim/ssim.m>

²<http://www.mathworks.com/matlabcentral/fileexchange/28689-hog-descriptor-for-matlab/content/HOG.m>

³http://www.mathworks.com/matlabcentral/fileexchange/30242-kinect-matlab/content/OpenNI1/Example/template_matching.m

el VPU-Lab. En el resto de los casos el código se ha extraído de la web de los autores donde se encuentra disponible.

5.3. *Dataset*

Para realizar la evaluación del algoritmo se ha utilizado el *dataset* público SOVTds (*Single Object Video Tracking dataset*) [13] creado por el VPU-Lab. Este *dataset* cubre los problemas más comunes en el seguimiento de objetos en diferentes situaciones (sintéticas y reales) y con distintos tipos de objetos (elipses sintéticas, coches, caras y personas). Los problemas que se incluyen en el *dataset* son:

- **Movimiento complejo.** El objeto cambia su trayectoria inesperadamente o incrementa su velocidad de manera abrupta, excediendo así el área de búsqueda del *tracker*.
- **Cambios de iluminación graduales.** Las secuencias pueden sufrir pequeños cambios en el valor medio de la intensidad del *frame*, modificando el modelo del objeto y haciendo que el seguimiento sea más difícil.
- **Cambios de iluminación abruptos.** Cuando el objeto se mueve puede entrar en áreas con iluminación variable. Esto puede confundir al *tracker* haciendo que pierda al objeto.
- **Ruido.** Aparece como variaciones aleatorias en los valores de los píxeles de las imágenes, pudiendo degradar significativamente la calidad de las características extraídas para el modelo del objeto.
- **Oclusión.** Se produce cuando un objeto se interpone entre la cámara y el objeto seguido ocultándolo total o parcialmente.
- **Cambios de escala.** Ocurre cuando el objeto aumenta o disminuye su tamaño debido a que varía su distancia a la cámara.
- **Objetos similares.** Son objetos con características similares a las del objeto seguido (color, bordes...), por lo que el *tracker* puede confundirlos y seguirlos.

Las secuencias que forman el *dataset* se dividen en cuatro niveles (ver Tabla 5.1). En la Figura 5.1 se muestran ejemplos de *frames* de cada uno de los niveles en los que aparecen diferentes problemas.

Para obtener una descripción detallada acerca de los factores utilizados para evaluar la complejidad de cada problema y de las situaciones modeladas en el *dataset* ver el Apéndice B.

Nivel	Nº de secuencias	Nº de <i>frames</i>	Problemas
S1 (secuencias sintéticas)	35	~3500	Movimiento complejo Cambios de iluminación globales Cambios de iluminación abruptos Ruido Oclusión Cambios de escala Objetos similares
S2 (secuencias de laboratorio)	21	~6500	
S3 (secuencias reales simples)	52	~8500	
S4 (secuencias reales complejas)	18	~4500	

Tabla 5.1: Niveles del *dataset* SOVTds. Para cada nivel se muestra el número total de secuencias que lo componen y el número aproximado de *frames*. En todos los niveles se modelan los problemas de seguimiento indicados en la última columna. En los niveles S1, S2 y S3 estos problemas aparecen de manera aislada en cada secuencia. En el nivel S4 hay una combinación de dos o más problemas.

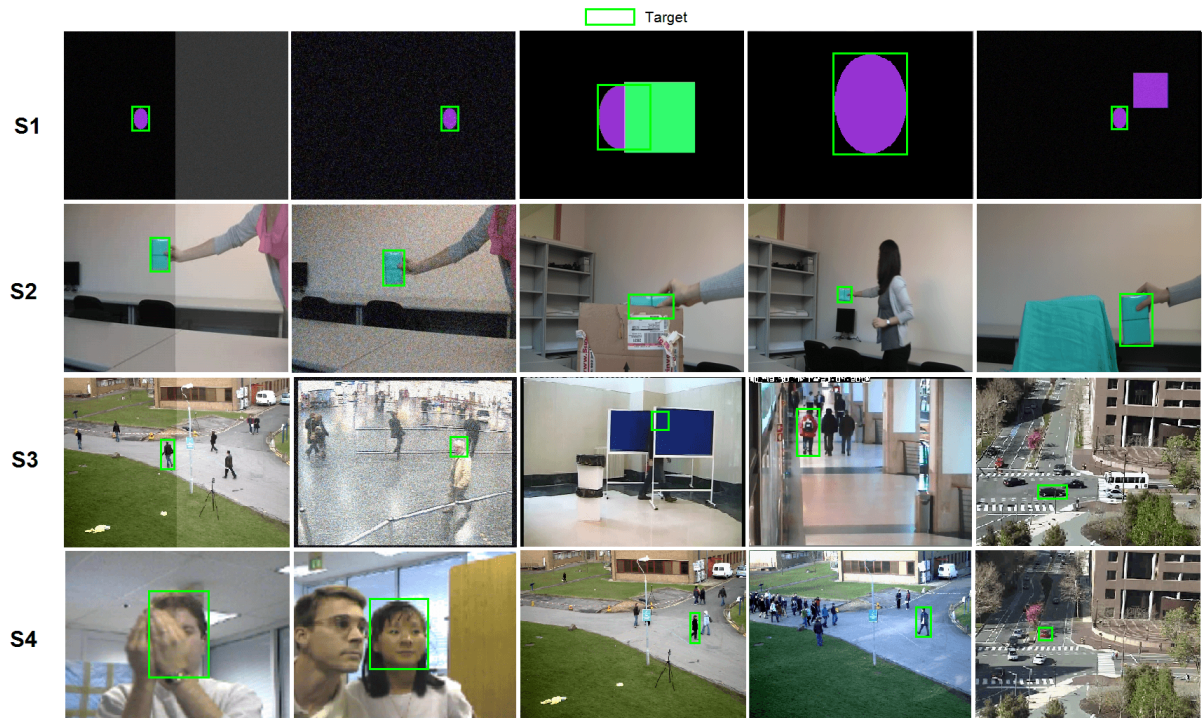


Figura 5.1: Ejemplos de *frames* pertenecientes a secuencias de cada uno de los niveles del *dataset* SOVTds. De arriba abajo: secuencias sintéticas (S1), secuencias de laboratorio (S2), secuencias reales simples (S3) y secuencias reales complejas (S4). Para los niveles S1, S2 y S3 se presentan, de izquierda a derecha, los siguientes problemas: cambios de iluminación abruptos, ruido, oclusión, cambios de escala y objetos similares. Para el nivel S4 se muestra, de izquierda a derecha, la combinación de los siguientes problemas: oclusión y objetos similares; cambios de escala y objetos similares; oclusión, cambios de escala y objetos similares; cambios de iluminación graduales, oclusión y cambios de escala; cambios de iluminación abruptos y cambios de escala.

5.4. Métricas

A continuación se definen las métricas usadas en la evaluación del algoritmo:

- En la etapa de detección de los *frames* de cambio se han utilizado las medidas estándar de *Precision* (P), *Recall* (R) y *F-score* (F):

$$P = \frac{TP}{TP + FP}, \quad (5.1)$$

$$R = \frac{TP}{TP + FN}, \quad (5.2)$$

$$F = \frac{2 \cdot P \cdot R}{P + R}, \quad (5.3)$$

donde TP indica el número de *frames* de cambio detectados correctamente, FP es el número de detecciones erróneas y FN hace referencia al número de *frames* de cambio no detectados. En el momento de realizar la contabilidad se han tenido en cuenta las posibles desviaciones del algoritmo, por lo que se han considerado ventanas de ± 5 *frames* de tolerancia en torno a los *frames* de cambio anotados. De esta manera todas las detecciones dentro de las ventanas se han contabilizado como un único TP.

Para anotar los *frames* de cambio que se pretenden detectar se ha usado la señal de error de GT:

$$e_t = 1 - SO_t, \quad (5.4)$$

donde SO_t es el solape espacial entre la imagen del objeto estimada por el *tracker* y la imagen obtenida a partir de las anotaciones de GT (ver Figura 5.2). Este solape se calcula usando el coeficiente de Dice (Fórmula 2.8). Cuando la señal de error de GT pasa de ser menor que uno a ser igual uno ($e_{t-1} < 1 \wedge e_t = 1$) se ha anotado un cambio CT→WT. Cuando sucede lo contrario ($e_{t-1} = 1 \wedge e_t < 1$) entonces se ha anotado un cambio WT→CT. La señal de cambio anotada queda entonces definida como sigue:

$$change_t = \begin{cases} 1 & \text{si } (e_{t-1} < 1 \wedge e_t = 1) \vee (e_{t-1} = 1 \wedge e_t < 1) \\ 0 & \text{resto} \end{cases} \quad (5.5)$$

- En la etapa de decisión CT/WT se han empleado las medidas *True Positive Rate* (TPR) y *False Positive Rate* (FPR):

$$TPR = \frac{TP}{TP + FN}, \quad (5.6)$$

$$FPR = \frac{FP}{FP + TN}, \quad (5.7)$$

donde en este caso TP y TN indican, respectivamente, el número de decisiones CT y WT

correctas y FP y FN denotan, respectivamente, el número de decisiones CT y WT erróneas. Además se ha hecho uso de la métrica *Area Under the Curve* (AUC) obtenida a partir de la curva ROC (*Receiver Operating Characteristic*), la cual muestra la relación entre TPR y FPR (ver Figura 5.3).

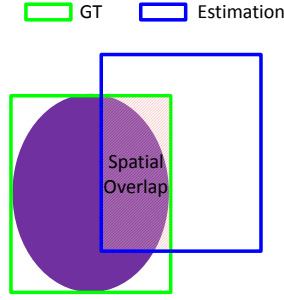


Figura 5.2: Ejemplo de solape espacial entre la imagen del objeto estimada por el *tracker* y la imagen obtenida a partir de las anotaciones de GT

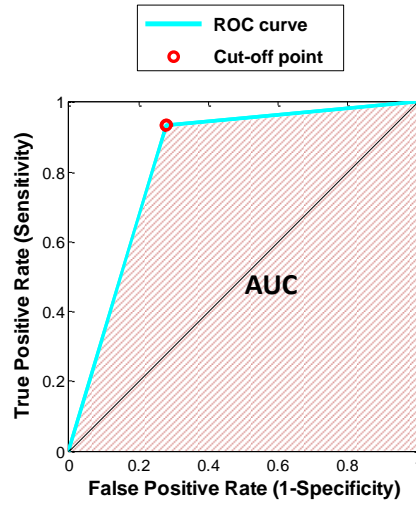


Figura 5.3: Ejemplo de curva ROC.

5.5. Análisis de las características

En los siguientes apartados se lleva a cabo un análisis de las características presentadas en el Capítulo 4. Para realizar dicho análisis se han empleado todas las secuencias del *dataset* (junto con las anotaciones de GT asociadas) y los datos de seguimiento de los seis *trackers*. Primero se evalúan diferentes modelos de seguimiento obtenidos a partir de las características (Sección 5.5.1). Después se estudia la correlación existente entre cada una de ellas (Sección 5.5.2).

5.5.1. Modelado

Para cada característica f se definen dos clases: CT para el seguimiento correcto y WT para el seguimiento incorrecto. De cada una se plantean distintos posibles modelos que se describen a continuación:

- **Clase CT.** Se considera, por una parte, un modelo creado a partir de los valores de la característica obtenidos usando las anotaciones de GT y, por otra, varios modelos generados con los valores de la característica obtenidos empleando los datos de seguimiento de los *trackers*. En este último caso se seleccionan aquellos valores correspondientes a los *frames* t en los que el solape espacial (SO_t) supere un umbral ξ :

$$\{f_t : SO_t > \xi\}, \quad (5.8)$$

con $\xi \in \{0.00, 0.25, 0.50, 0.75\}$. El objetivo es determinar cuál de los modelos obtenidos con los datos de seguimiento es el más adecuado para posteriormente utilizarlo en el algoritmo.

- **Clase WT.** Se considera un único modelo creado a partir de los valores de la característica obtenidos usando los datos de seguimiento. En esta ocasión se toman los valores relativos a los *frames* t en los que $SO_t = 0$:

$$\{f_t : SO_t = 0\} \quad (5.9)$$

El propósito es demostrar que esta clase es similar a la clase CT.

En primer lugar se mide la similitud entre los distintos modelos de CT y el modelo de WT. Para realizar esta comparación se ha empleado la distancia de Bhattacharyya [53] entre los histogramas normalizados $\hat{\mathbf{H}}_{CT}$ y $\hat{\mathbf{H}}_{WT}$ asociados a cada uno de los modelos:

$$d(\hat{\mathbf{H}}_{CT}, \hat{\mathbf{H}}_{WT}) = \sqrt{1 - \rho(\hat{\mathbf{H}}_{CT}, \hat{\mathbf{H}}_{WT})}, \quad (5.10)$$

donde:

$$\rho(\hat{\mathbf{H}}_{CT}, \hat{\mathbf{H}}_{WT}) = \sum_{i=1}^n \sqrt{\hat{\mathbf{H}}_{CT}(i) \cdot \hat{\mathbf{H}}_{WT}(i)} \quad (5.11)$$

es el coeficiente de Bhattacharyya calculado entre los n bins de los histogramas normalizados. Los cálculos se han realizado a partir de las secuencias del *dataset* completo. Para generar los diferentes modelos se han utilizado los datos de seguimiento de todos los *trackers* y las anotaciones de GT. En la Tabla 5.2 se recogen los resultados obtenidos con cada una de las características. Se puede apreciar que las distancias son bajas en todos los casos, lo que indica una elevada

Característica	$d(\hat{\mathbf{H}}_{CT}, \hat{\mathbf{H}}_{WT})$				
	CT (GT)	CT (TRACKING)			
		(SO _t > 0.00)	(SO _t > 0.25)	(SO _t > 0.50)	(SO _t > 0.75)
<i>shape ratio score</i>	0.277	0.141	0.135	0.140	0.157
<i>area ratio score</i>	0.309	0.142	0.159	0.166	0.162
<i>direction smoothness</i>	0.344	0.234	0.242	0.252	0.264
<i>speed smoothness</i>	0.255	0.263	0.268	0.280	0.289
<i>histogram difference red</i>	0.287	0.240	0.247	0.330	0.348
<i>histogram difference green</i>	0.368	0.229	0.235	0.257	0.295
<i>histogram difference blue</i>	0.433	0.238	0.245	0.281	0.327
<i>histogram difference grayscale</i>	0.342	0.180	0.191	0.211	0.293
<i>SSD</i>	0.149	0.079	0.083	0.086	0.097
<i>texture difference</i>	0.140	0.151	0.164	0.171	0.233
<i>DSSIM</i>	0.277	0.202	0.227	0.241	0.262
<i>HOG difference</i>	0.330	0.251	0.261	0.275	0.298
<i>CLD difference</i>	0.369	0.215	0.226	0.237	0.247
<i>covariance matrix difference</i>	0.445	0.341	0.353	0.360	0.425

Tabla 5.2: Distancia de Bhattacharyya entre los histogramas de los distintos modelos de CT (GT y datos de seguimiento con $SO_t > \xi$) y el histograma del modelo de WT (datos de seguimiento con $SO_t = 0$). Para generar dichos modelos se han usado las anotaciones de GT y los datos de seguimiento de todos los *trackers*. A medida que los valores de la distancia se aproximan a 0 indican un parecido más alto.

Característica	$d(\hat{\mathbf{H}}_{CT(TRACKING)}, \hat{\mathbf{H}}_{CT(GT)})$			
	CT (TRACKING)			
	(SO _t > 0.00)	(SO _t > 0.25)	(SO _t > 0.50)	(SO _t > 0.75)
<i>shape ratio score</i>	0.319	0.320	0.322	0.333
<i>area ratio score</i>	0.372	0.380	0.382	0.381
<i>direction smoothness</i>	0.355	0.362	0.367	0.389
<i>speed smoothness</i>	0.271	0.276	0.282	0.284
<i>histogram difference red</i>	0.104	0.102	0.120	0.142
<i>histogram difference green</i>	0.286	0.288	0.249	0.196
<i>histogram difference blue</i>	0.337	0.338	0.289	0.241
<i>histogram difference grayscale</i>	0.298	0.298	0.260	0.159
<i>SSD</i>	0.140	0.137	0.142	0.153
<i>texture difference</i>	0.051	0.057	0.062	0.123
<i>DSSIM</i>	0.233	0.202	0.201	0.171
<i>HOG difference</i>	0.360	0.270	0.278	0.106
<i>CLD difference</i>	0.422	0.429	0.435	0.359
<i>covariance matrix difference</i>	0.238	0.257	0.269	0.142

Tabla 5.3: Distancia de Bhattacharyya entre los histogramas de los modelos de CT obtenidos a partir de los datos de seguimiento de todos los *trackers* y el histograma del modelo de CT obtenido a partir de los datos de GT. A medida que los valores de la distancia se aproximan a 0 indican un parecido más alto.

similitud entre todos los modelos de CT y el modelo de WT. Es decir, los valores que toman las características cuando el seguimiento es correcto y cuando el seguimiento es incorrecto son muy parecidos. Por tanto *no se pueden considerar dos clases distintas, puesto que no hay una clara discriminación entre CT y WT*. Se verifica así que el uso de las características en [11] no es el adecuado.

Por otro lado también se lleva a cabo una comparación entre los modelos de CT obtenidos a partir de los datos de seguimiento de todos los *trackers* y el modelo de CT generado utilizando las anotaciones de GT. Los resultados se reflejan en la Tabla 5.3. Se comprueba que, en general, todos los modelos creados con los datos de seguimiento se asemejan bastante al modelo obtenido a partir las anotaciones de GT, por lo que cualquiera de ellos proporcionará unos resultados parecidos. En este trabajo se utilizará el modelo con $SO_t > 0.00$.

Para el modelo de CT escogido se busca la similitud entre las distribuciones reales de cada característica y modelos de distribución estándar (Normal, Beta, Gamma, Exponencial y Rayleigh). El propósito es ajustar las distribuciones obtenidas empíricamente a distribuciones teóricas, que serán las que se utilicen en el algoritmo. Para ello se usa el test de Kolmogórov-Smirnov [60], que indica la bondad de ajuste entre dos distribuciones de probabilidad. En la Tabla 5.4 se muestran los resultados del test. Cada característica se modelará con la distribución que ha obtenido el estadístico más bajo.

Característica	Distribución				
	Normal	Beta*	Gamma	Exponencial	Rayleigh
<i>shape ratio score</i>	0.371	0.290	0.699	0.699	0.730
<i>area ratio score</i>	0.361	0.292	0.693	0.693	0.712
<i>direction smoothness</i>	0.176	0.546	0.347	0.313	0.387
<i>speed smoothness</i>	0.163	0.398	0.180	0.209	0.157
<i>histogram difference red</i>	0.167	0.721	0.272	0.129	0.332
<i>histogram difference green</i>	0.167	0.722	0.270	0.127	0.331
<i>histogram difference blue</i>	0.167	0.717	0.274	0.135	0.330
<i>histogram difference grayscale</i>	0.093	0.713	0.327	0.157	0.186
<i>SSD</i>	0.368	-	0.144	0.388	0.662
<i>texture difference</i>	0.349	-	0.177	0.369	0.673
<i>DSSIM</i>	0.185	0.722	0.220	0.194	0.364
<i>HOG difference</i>	0.180	0.761	0.350	0.075	0.346
<i>CLD difference</i>	0.233	-	0.365	0.107	0.430
<i>covariance matrix difference</i>	0.277	-	0.338	0.141	0.530

Tabla 5.4: Estadísticos obtenidos para cada característica en el test de Kolmogórov-Smirnov. En negrita se muestran los mejores resultados.

* El dominio de la distribución Beta es $[0, 1]$, por lo que no es válida para *SSD*, *texture difference*, *CLD difference* y *covariance matrix difference*.

5.5.2. Correlación

En este apartado se analiza el grado de dependencia existente entre cada una de las características. Para ello se ha utilizado el coeficiente de correlación de Pearson [61], que mide la relación lineal entre dos variables aleatorias \mathbf{X} e \mathbf{Y} . Considerando dos muestras $\{X_1, \dots, X_n\}$ y $\{Y_1, \dots, Y_n\}$, el valor r de dicho coeficiente se estima a partir de la siguiente fórmula:

$$r = \frac{\sum_{i=1}^n (X_i - \bar{X}) (Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}}, \quad (5.12)$$

donde \bar{X} e \bar{Y} denotan los valores medios de sendas muestras y, n , la dimensión de las mismas.

Los cálculos de la correlación se han llevado a cabo empleando los valores de las características obtenidos a partir de los datos de GT del *dataset* completo. En la Tabla 5.5 se muestra el valor absoluto del coeficiente de correlación obtenido entre cada una de las características. Observando los resultados se puede comprobar que existe una fuerte correlación entre las características de diferencia de histograma. En el caso de *histogram difference red*, *histogram difference green* e *histogram difference blue* la dependencia es casi total. Esto indica que el hecho de analizar por separado los tres canales RGB es redundante. También se aprecia que hay una alta correlación entre *shape ratio score* y *area ratio score*, ya que ambas dependen del tamaño del *bounding box*. Cabe destacar además la elevada dependencia entre *CLD difference* y *DSSIM* y entre *CLD difference* y *HOG difference*.

	<i>shape ratio score</i>	<i>area ratio score</i>	<i>direct. smooth.</i>	<i>speed smooth.</i>	<i>histog. differ. red</i>	<i>histog. differ. green</i>	<i>histog. differ. blue</i>	<i>histog. differ. grayscale</i>	<i>SSD</i>	<i>texture differ.</i>	<i>DSSIM</i>	<i>HOG differ.</i>	<i>CLD differ.</i>	<i>covar. matrix differ.</i>
<i>shape ratio score</i>	1.000	0.750	0.061	0.061	0.017	0.014	0.002	0.002	0.175	0.006	0.024	0.201	0.122	0.122
<i>area ratio score</i>		1.000	0.151	0.070	0.023	0.024	0.005	0.013	0.208	0.014	0.040	0.200	0.139	0.125
<i>direct. smooth.</i>			1.000	0.180	0.067	0.088	0.071	0.057	0.082	0.031	0.188	0.151	0.243	0.087
<i>speed smooth.</i>				1.000	0.141	0.206	0.141	0.194	0.014	0.039	0.131	0.032	0.034	0.026
<i>histog. differ. red</i>					1.000	<i>0.934</i>	<i>0.919</i>	<i>0.652</i>	0.081	0.080	0.357	0.252	0.280	0.255
<i>histog. differ. green</i>						1.000	<i>0.931</i>	<i>0.687</i>	0.090	0.092	0.305	0.217	0.280	0.237
<i>histog. differ. blue</i>							1.000	<i>0.639</i>	0.065	0.066	0.387	0.220	0.280	0.242
<i>histog. differ. grayscale</i>								1.000	0.215	0.277	0.466	0.303	0.455	0.222
<i>SSD</i>									1.000	0.262	0.264	0.277	0.341	0.169
<i>texture differ.</i>										1.000	0.310	0.233	0.473	0.140
<i>DSSIM</i>											1.000	0.414	<i>0.716</i>	0.206
<i>HOG differ.</i>												1.000	<i>0.646</i>	<i>0.546</i>
<i>CLD differ.</i>													1.000	0.374
<i>covar. matrix differ.</i>														1.000
Media	0.120	0.135	0.112	0.098	0.312	0.316	0.305	0.322	0.156	0.173	0.293	0.284	0.337	0.212

Tabla 5.5: Coeficiente de correlación de Pearson (en valor absoluto) entre cada una de las características. En negrita se resaltan los valores bajos y en cursiva los valores altos. Al ser el coeficiente de correlación una medida simétrica se han omitido los valores por debajo de la diagonal.

5.6. Evaluación del algoritmo propuesto

A continuación se presentan los resultados conseguidos con el algoritmo propuesto utilizando la estrategia de detección de anomalías (Sección 5.6.1). Además se proporciona un resumen de los resultados obtenidos usando las técnicas de clasificación supervisada planteadas en el Anexo A como alternativa a la detección de anomalías (Sección 5.6.2).

5.6.1. Evaluación usando detección de anomalías

Para llevar a cabo la evaluación se han empleado, en cada *tracker*, 76 secuencias de entrenamiento, 25 de validación cruzada y 25 de test seleccionadas de manera aleatoria. En la máquina de estados se ha usado *covariance matrix difference* para realizar la comparación con la imagen del objeto en el *frame* inicial. Se ha escogido esta característica porque es la que ha tenido un mejor funcionamiento.

En la Tabla 5.6 se exponen los resultados obtenidos utilizando *un modelo de CT por cada característica*. Se puede apreciar que los valores de F que consiguen en la etapa de detección la mayor parte de las características son demasiado bajos. La mejor puntuación la obtiene, en todos los casos, *covariance matrix difference*, alcanzando unos valores aceptables en MS y TLD. A pesar del bajo F, algunas de las características consiguen unos resultados mejores en la etapa de decisión debido a que la máquina de estados ignora muchas de las falsas detecciones. En general, esto sucede con las características que han obtenido un alto R. En las Figuras 5.4, 5.5 y 5.6 se muestran ejemplos en los que el algoritmo ha funcionado correctamente usando *covariance matrix difference*, *HOG difference* y *CLD difference*. En las Figuras 5.7 y 5.8 también se incluyen ejemplos con *shape ratio score* y *texture difference* en los que no se han logrado detectar los *frames* de cambio y, como consecuencia, no se han conseguido identificar los segmentos temporales WT.

					Característica													
					shape	area			histog.	histog.	histog.	histog.						
					ratio	ratio	direct.	speed	histog.	histog.	histog.	histog.	SSD	texture	DSSIM	HOG	CLD	covar.
					score	score	smooth	smooth	red	green	blue	grayscale		differ.		differ.	differ.	differ.
Tracker	MS*	Detección frames de cambio	P	-	-	0.161	0.053	1.000	0.095	1.000	0.391	0.286	0.077	0.184	0.461	0.741	0.688	
			R	-	-	0.628	0.233	0.070	0.279	0.140	0.209	0.093	0.047	0.163	0.814	0.465	0.767	
			F	-	-	0.256	0.086	0.130	0.142	0.245	0.273	0.140	0.058	0.173	0.588	0.571	0.725	
		Decisión CT/WT	TPR	-	-	0.927	0.926	1.000	0.950	0.993	0.999	0.999	0.999	0.947	0.950	0.984	0.976	
			FPR	-	-	0.221	0.604	1.000	0.872	0.954	0.819	0.966	0.816	0.573	0.267	0.665	0.270	
			AUC	-	-	0.853	0.661	0.500	0.539	0.520	0.590	0.517	0.592	0.687	0.842	0.659	0.853	
	CPF	Detección frames de cambio	P	0.093	0.077	0.077	0.050	0.615	0.667	0.579	0.750	1.000	0.163	0.160	0.234	0.547	0.659	
			R	0.254	0.349	0.318	0.571	0.127	0.064	0.175	0.143	0.032	0.349	0.381	0.476	0.460	0.460	
			F	0.136	0.126	0.124	0.092	0.211	0.116	0.268	0.240	0.062	0.222	0.225	0.313	0.500	0.542	
		Decisión CT/WT	TPR	0.785	0.731	0.837	0.783	0.886	0.903	0.801	0.880	0.991	0.902	0.897	0.845	0.774	0.814	
			FPR	0.438	0.240	0.272	0.137	0.679	0.692	0.750	0.552	1.000	0.607	0.288	0.398	0.612	0.384	
			AUC	0.674	0.746	0.783	0.823	0.604	0.606	0.526	0.664	0.496	0.648	0.805	0.724	0.581	0.715	
	LOT	Detección frames de cambio	P	0.200	0.140	0.032	0.037	0.057	0.036	0.078	0.333	0.027	-	0.500	0.143	0.150	0.181	
			R	0.065	0.258	0.290	0.710	0.387	0.355	0.161	0.129	0.065	0.000	0.032	0.097	0.097	0.419	
			F	0.098	0.182	0.057	0.071	0.099	0.066	0.105	0.186	0.038	-	0.061	0.115	0.118	0.252	
		Decisión CT/WT	TPR	0.998	0.976	0.921	0.929	0.951	0.950	0.972	0.986	0.970	1.000	0.998	0.998	0.997	0.974	
			FPR	0.971	0.537	0.308	0.276	0.565	0.562	0.694	0.892	0.874	1.000	0.986	0.907	0.931	0.511	
			AUC	0.514	0.720	0.807	0.827	0.693	0.694	0.639	0.547	0.548	0.500	0.506	0.546	0.533	0.732	
	IVT	Detección frames de cambio	P	0.049	0.044	0.030	0.047	0.085	0.282	0.022	0.048	0.077	0.023	0.135	0.037	0.054	0.290	
			R	0.320	0.640	0.120	0.200	0.200	0.080	0.160	0.160	0.040	0.200	0.200	0.120	0.480	0.360	
			F	0.086	0.082	0.048	0.076	0.119	0.051	0.039	0.073	0.053	0.041	0.161	0.056	0.098	0.321	
		Decisión CT/WT	TPR	0.979	0.941	0.850	0.859	0.994	0.999	0.911	0.999	0.987	0.984	0.996	0.999	0.916	0.983	
			FPR	0.290	0.169	0.301	0.556	0.509	0.955	0.823	0.873	0.867	0.644	0.885	0.680	0.196	0.304	
			AUC	0.845	0.886	0.775	0.652	0.743	0.522	0.544	0.563	0.560	0.670	0.555	0.660	0.860	0.840	
SPT*	Detección frames de cambio	P	-	-	0.000	0.000	1.000	1.000	0.042	1.000	0.014	0.028	0.093	0.409	0.250	0.333		
		R	-	-	0.000	0.000	0.029	0.029	0.114	0.057	0.371	0.114	0.257	0.257	0.200	0.514		
		F	-	-	-	-	0.056	0.056	0.062	0.108	0.027	0.044	0.136	0.316	0.222	0.405		
	Decisión CT/WT	TPR	-	-	0.993	0.981	0.992	1.000	0.979	1.000	0.887	0.956	0.977	0.988	0.967	0.939		
		FPR	-	-	0.649	0.826	0.870	1.000	0.724	0.966	0.442	0.677	0.873	0.588	0.766	0.450		
		AUC	-	-	0.672	0.578	0.561	0.500	0.628	0.517	0.723	0.640	0.552	0.700	0.601	0.745		
TLD	Detección frames de cambio	P	0.086	0.375	0.160	0.047	0.667	0.667	0.636	0.700	0.116	0.122	0.077	0.500	0.750	0.618		
		R	0.068	0.341	0.091	0.818	0.181	0.136	0.159	0.159	0.296	0.273	0.296	0.432	0.477	0.773		
		F	0.076	0.357	0.116	0.090	0.286	0.226	0.255	0.259	0.167	0.169	0.123	0.463	0.583	0.687		
	Decisión CT/WT	TPR	0.954	0.970	0.889	0.834	0.980	1.000	0.987	0.999	0.976	0.960	0.977	0.967	0.981	0.926		
		FPR	0.647	0.700	0.954	0.174	0.889	0.904	0.685	0.904	0.853	0.658	0.295	0.267	0.606	0.236		
		AUC	0.654	0.635	0.468	0.830	0.546	0.548	0.651	0.548	0.562	0.651	0.841	0.850	0.688	0.845		
Media	Detección frames de cambio	P	0.107	0.159	0.077	0.039	0.571	0.458	0.393	0.537	0.253	0.069	0.192	0.297	0.415	0.462		
		R	0.177	0.397	0.241	0.422	0.166	0.157	0.152	0.143	0.150	0.164	0.222	0.367	0.363	0.549		
		F	0.099	0.187	0.100	0.069	0.150	0.110	0.162	0.190	0.081	0.089	0.147	0.309	0.349	0.489		
	Decisión CT/WT	TPR	0.929	0.905	0.903	0.885	0.967	0.967	0.941	0.977	0.968	0.967	0.965	0.958	0.937	0.935		
		FPR	0.587	0.412	0.451	0.429	0.752	0.831	0.772	0.834	0.834	0.734	0.650	0.518	0.629	0.359		
		AUC	0.672	0.747	0.726	0.729	0.608	0.568	0.585	0.572	0.568	0.617	0.658	0.720	0.654	0.788		

Tabla 5.6: Resultados obtenidos con el algoritmo propuesto usando *un modelo de CT por cada característica*. Los tres valores de F más altos se resaltan en negrita (1º), azul (2º) y rojo (3º).

* En MS y SPT no se han utilizado las características *shape ratio score* y *area ratio score* ya que el tamaño del *bounding box* en ambos *trackers* es constante.

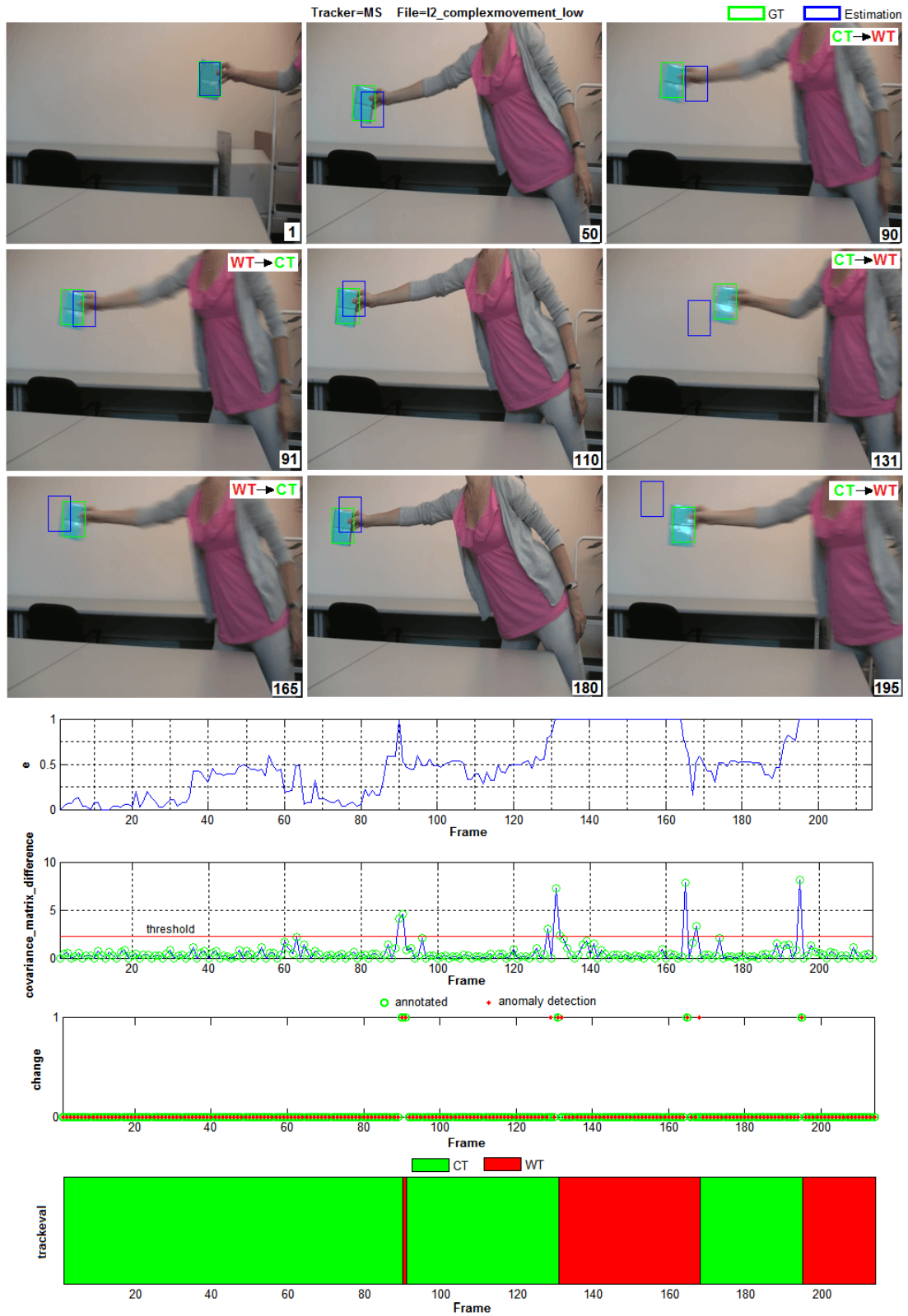


Figura 5.4: Ejemplo de funcionamiento del algoritmo propuesto utilizando *covariance matrix difference* sobre MS. Arriba se muestran los *frames* 1, 50, 90, 91, 110, 131, 165, 180 y 195 de la secuencia. En las gráficas de abajo se representa la señal de error de GT, el valor de la característica en cada *frame*, la señal de cambio anotada junto con la obtenida con el algoritmo y las decisiones de la máquina de estados en cada *frame*.

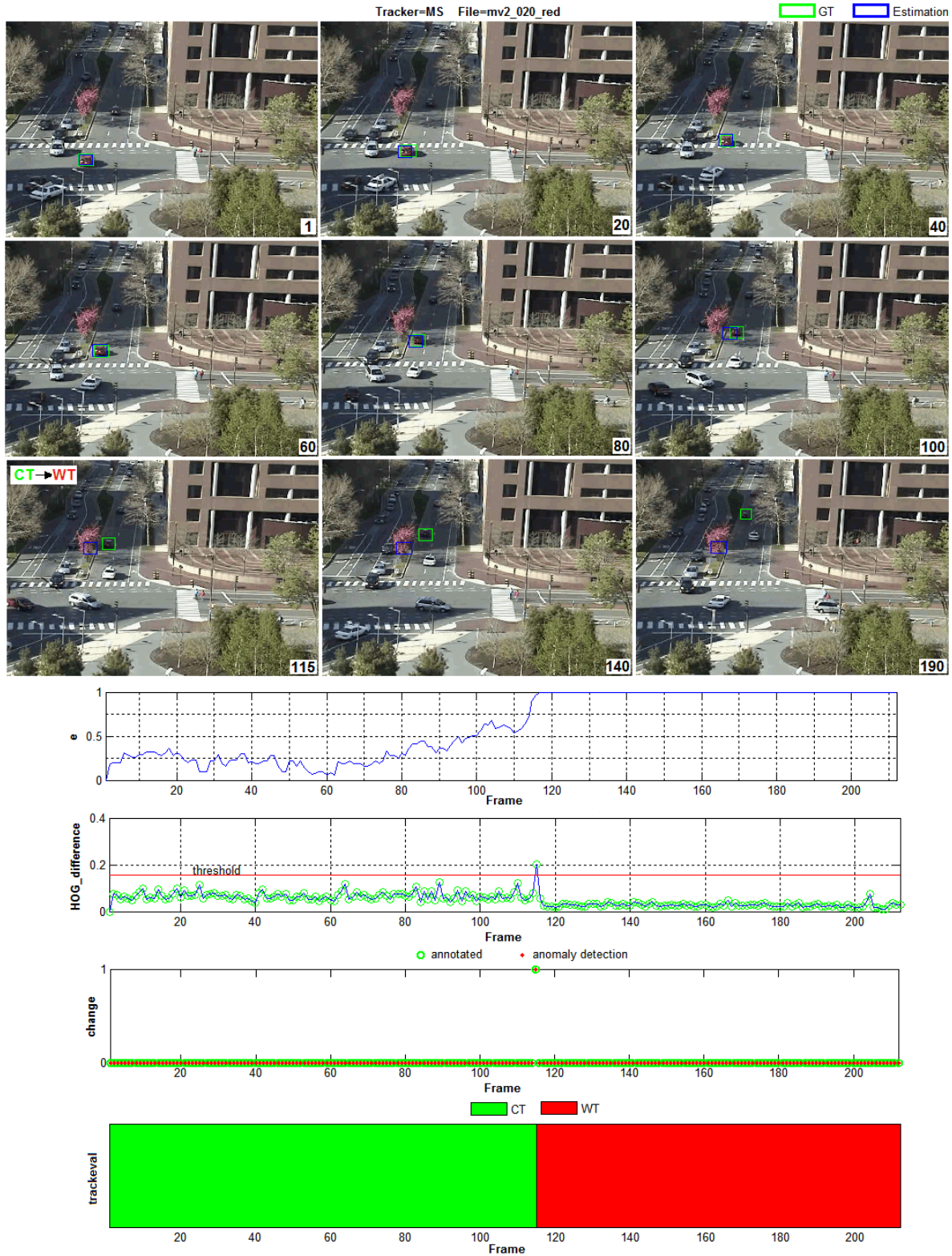


Figura 5.5: Ejemplo de funcionamiento del algoritmo propuesto utilizando *HOG difference* sobre MS. Arriba se muestran los *frames* 1, 20, 40, 60, 80, 100, 115, 140 y 190 de la secuencia. En las gráficas de abajo se representa la señal de error de GT, el valor de la característica en cada *frame*, la señal de cambio anotada junto con la obtenida con el algoritmo y las decisiones de la máquina de estados en cada *frame*.

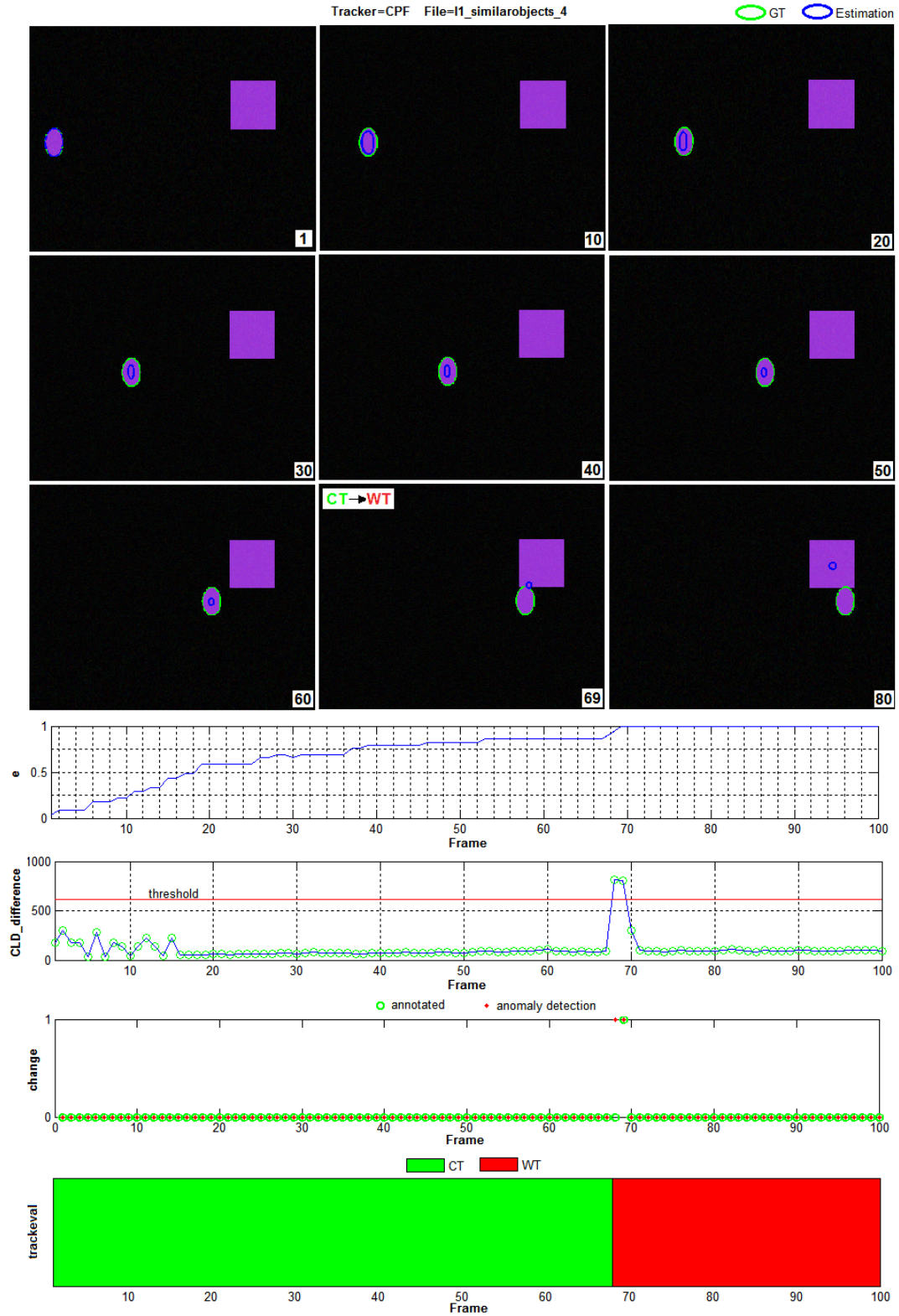


Figura 5.6: Ejemplo de funcionamiento del algoritmo propuesto utilizando *CLD difference* sobre CPF. Arriba se muestran los *frames* 1, 10, 20, 30, 40, 50, 60, 69 y 80 de la secuencia. En las gráficas de abajo se representa la señal de error de GT, el valor de la característica en cada *frame*, la señal de cambio anotada junto con la obtenida con el algoritmo y las decisiones de la máquina de estados en cada *frame*.

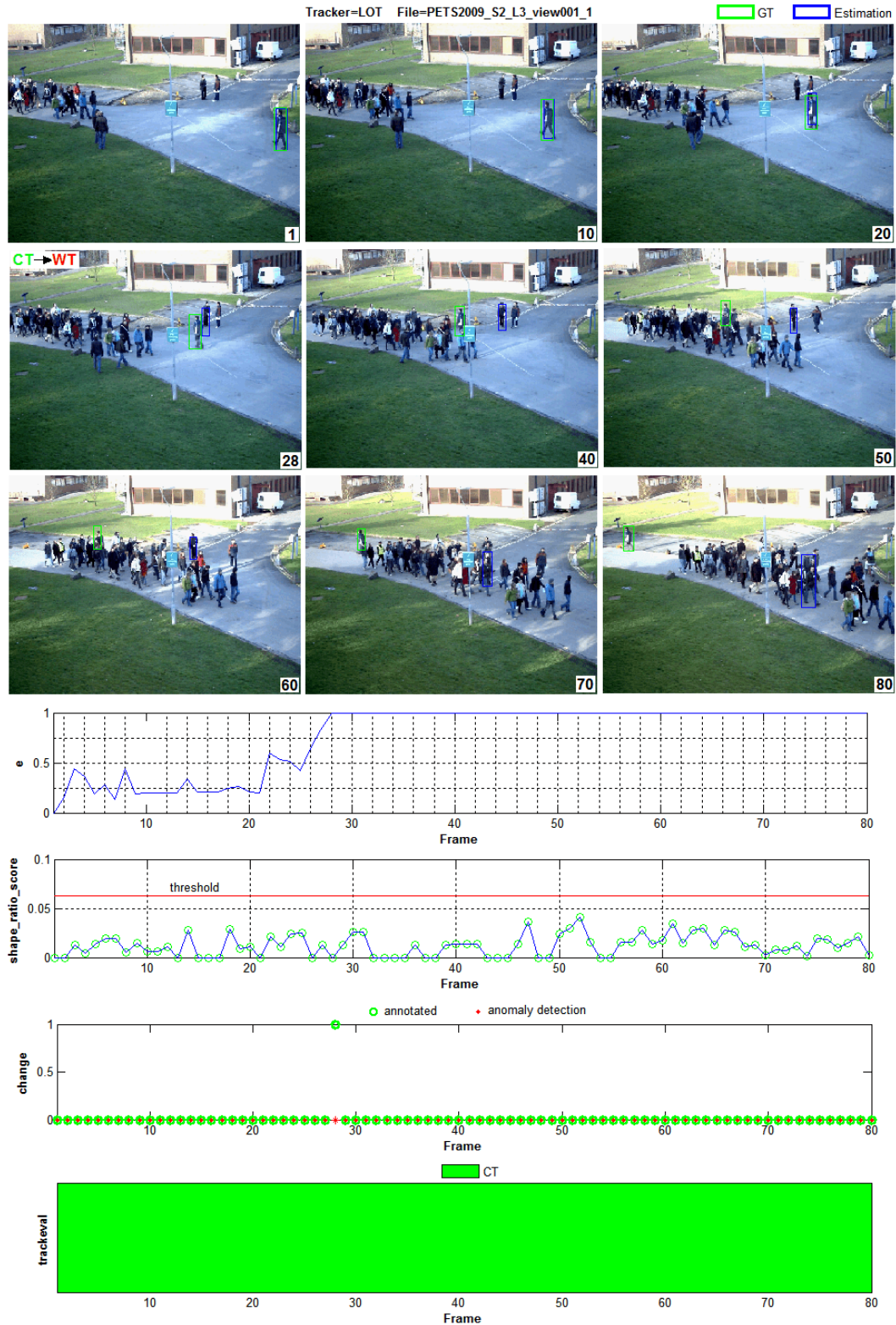


Figura 5.7: Ejemplo de funcionamiento del algoritmo propuesto utilizando *shape ratio score* sobre LOT. Arriba se muestran los *frames* 1, 10, 20, 28, 40, 50, 60, 70 y 80 de la secuencia. En las gráficas de abajo se representa la señal de error de GT, el valor de la característica en cada *frame*, la señal de cambio anotada junto con la obtenida con el algoritmo y las decisiones de la máquina de estados en cada *frame*.

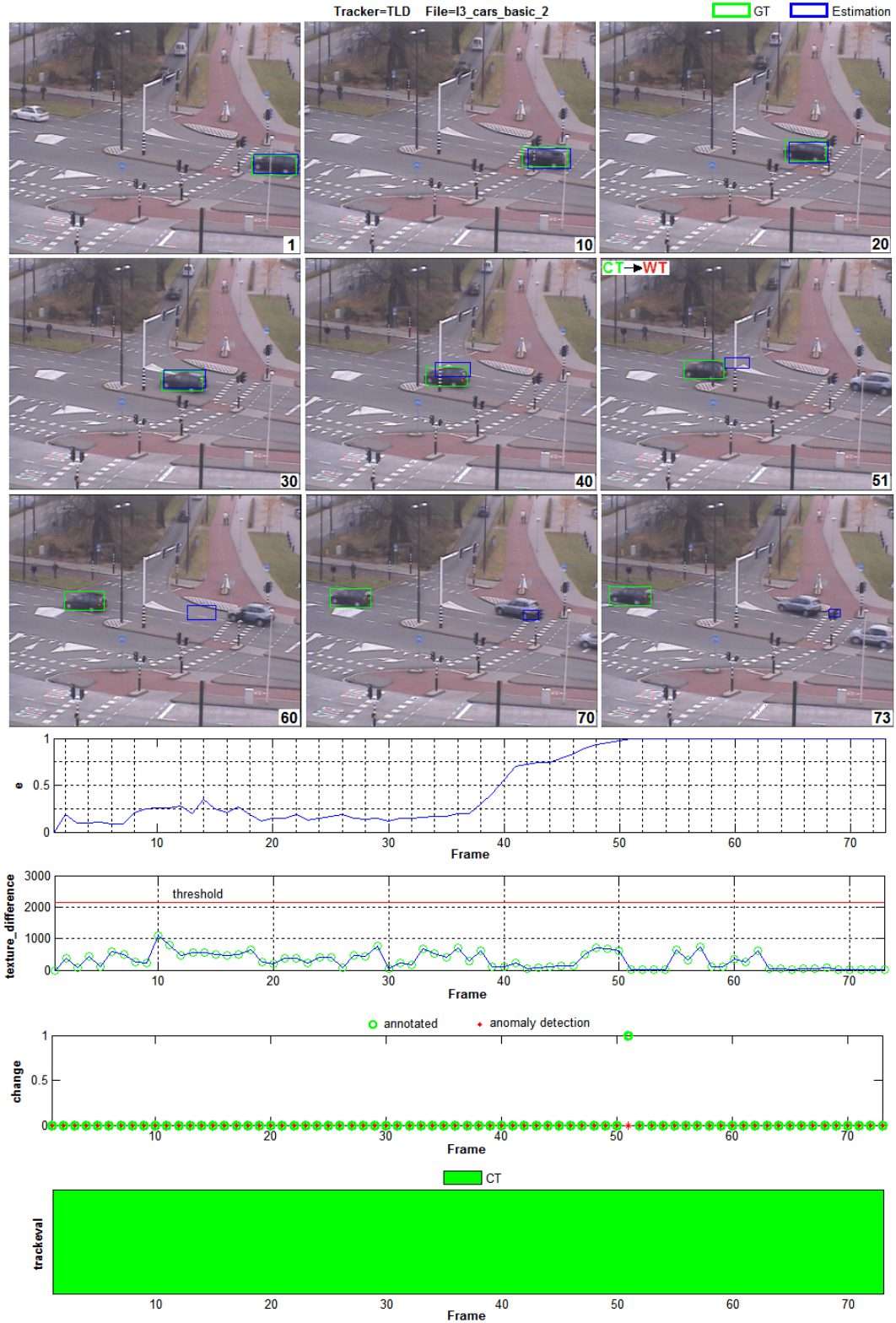


Figura 5.8: Ejemplo de funcionamiento del algoritmo propuesto utilizando *texture difference* sobre TLD. Arriba se muestran los *frames* 1, 10, 20, 30, 40, 51, 60, 70 y 73 de la secuencia. En las gráficas de abajo se representa la señal de error de GT, el valor de la característica en cada *frame*, la señal de cambio anotada junto con la obtenida con el algoritmo y las decisiones de la máquina de estados en cada *frame*.

		Características usadas	Detección <i>frames</i> de cambio			Decisión CT/WT		
			P	R	F	TPR	FPR	AUC
Tracker	MS	<i>covariance matrix difference</i>	0.897	0.605	0.722	0.964	0.606	0.679
		<i>HOG difference</i>						
	CPF	<i>covariance matrix difference</i>	0.635	0.746	0.686	0.818	0.481	0.669
		<i>CLD difference</i>						
	LOT	<i>covariance matrix difference</i>	0.344	0.355	0.349	0.978	0.552	0.713
		<i>histogram difference grayscale</i>						
	IVT	<i>covariance matrix difference</i>	0.310	0.360	0.333	0.983	0.304	0.840
		<i>DSSIM</i>						
	SPT	<i>covariance matrix difference</i>	0.424	0.400	0.412	0.950	0.530	0.710
		<i>HOG difference</i>						
	TLD	<i>covariance matrix difference</i>	0.667	0.773	0.716	0.911	0.224	0.844
		<i>CLD difference</i>						
Media			0.546	0.540	0.536	0.934	0.450	0.743

Tabla 5.7: Resultados obtenidos con el algoritmo propuesto usando *un único modelo de CT multivariante*. Para generar dicho modelo, en cada *tracker*, se han utilizado las dos características que individualmente han conseguido las mejores puntuaciones. Los tres valores de F más altos se resaltan en negrita (1º), azul (2º) y rojo (3º).

Como los resultados obtenidos por las características de manera individual no fueron muy satisfactorios, en la segunda versión propuesta se ha utilizado únicamente *covariance matrix difference* junto con la característica que ha logrado la segunda puntuación más alta. Por tanto, para cada *tracker*, se ha generado un modelo multivariante de CT a partir de las dos mejores características. En la Tabla 5.7 se recogen los resultados conseguidos con esta versión. Se aprecia, en casi todos los casos, una ligera mejora de F con respecto a los resultados obtenidos individualmente por *covariance matrix difference*. Sin embargo, en la etapa de decisión se comprueba que el valor de AUC ha disminuido en algunos *trackers*.

En la tercera versión se han combinado, en cada *tracker*, las detecciones de las dos características con mayor F. Se ha escogido, para el modelo de CT de cada característica, un umbral $\varepsilon_{opt(3)}^{(i)}$ que proporcione $P_{min} = 0.7$. Los resultados obtenidos se reflejan en la Tabla 5.8. Se puede comprobar que en MS, CPF y TLD el valor de F que se consigue tras la combinación es prácticamente el mismo que se obtiene de manera individual con *covariance matrix difference*. Esto indica que la segunda característica no ha aportado ninguna mejora. También se observa que en SPT ha bajado notablemente el valor de F debido a que R es cercano a 0. Lo que ha ocurrido es que al intentar conseguir una P alta se ha reducido mucho el valor de R. En el caso de LOT e IVT no ha sido posible lograr una P tan elevada con ninguna de las dos características, por lo que no ha habido ninguna detección.

		Características usadas	Detección <i>frames</i> de cambio			Decisión CT/WT		
			P	R	F	TPR	FPR	AUC
Tracker	MS	<i>covariance matrix difference</i>	0.824	0.651	0.727	0.967	0.402	0.783
		<i>HOG difference</i>						
	CPF	<i>covariance matrix difference</i>	0.743	0.413	0.531	0.782	0.544	0.619
		<i>CLD difference</i>						
	LOT	<i>covariance matrix difference</i>	-	0.000	-	1.000	1.000	0.500
		<i>histogram difference grayscale</i>						
	IVT	<i>covariance matrix difference</i>	-	0.000	-	1.000	1.000	0.500
		<i>DSSIM</i>						
	SPT	<i>covariance matrix difference</i>	0.667	0.057	0.105	0.999	0.957	0.521
		<i>HOG difference</i>						
	TLD	<i>covariance matrix difference</i>	0.667	0.682	0.674	0.966	0.282	0.842
		<i>CLD difference</i>						
	Media			0.484	0.301	0.340	0.952	0.698

Tabla 5.8: Resultados obtenidos con el algoritmo propuesto *combinando las detecciones de las características*. Se han empleado, en cada *tracker*, las dos características que individualmente han conseguido las puntuaciones más altas. Los tres valores de F más altos se resaltan en negrita (1º), azul (2º) y rojo (3º).

5.6.2. Evaluación usando clasificación supervisada

En este apartado se ofrece un extracto de los resultados conseguidos con el algoritmo propuesto empleando las técnicas de clasificación supervisada para detectar los *frames* de cambio. Los resultados detallados se encuentran en la Sección A.1.1 del Apéndice A.

En la Tabla 5.9 se muestra el valor medio de los resultados obtenidos con los distintos clasificadores débiles. En la Tabla 5.10 se presenta el valor medio de los resultados conseguidos con los diferentes métodos de combinación de clasificadores débiles. Se aprecia en todos los casos que los resultados son algo peores que los obtenidos con la estrategia de detección de anomalías. Esto es debido a que la clasificación supervisada es más apropiada cuando existe una gran cantidad de datos de entrenamiento de cada una de las clases. Sin embargo, en el algoritmo planteado en este trabajo los *frames* de cambio son muy pocos en comparación con el resto de *frames*. Por tanto, el número de vectores de características relativos a los *frames* de cambio es escaso y el modelo que genera el clasificador no es muy preciso. Además hay que considerar que cada clasificador sigue un procedimiento distinto para crear el modelo, mientras que la técnica de detección de anomalías utiliza un modelo adaptado para cada característica.

			Clasificador					Detección de anomalías (covar. matriz differ.)
			<i>Naive Bayes</i>	<i>Nearest Neighbors</i>	<i>Support Vector Machine</i>	<i>Classification Trees</i>	<i>Regression Trees</i>	
Media	Detección frames de cambio	P	0.709	0.428	0.413	0.535	0.388	0.462
		R	0.361	0.242	0.390	0.268	0.217	0.549
		F	0.397	0.304	0.304	0.340	0.269	0.489
	Decisión CT/WT	TPR	0.938	0.935	0.940	0.948	0.944	0.935
		FPR	0.599	0.537	0.532	0.599	0.520	0.359
		AUC	0.670	0.699	0.704	0.675	0.712	0.788

Tabla 5.9: Valor medio de los resultados obtenidos con el algoritmo propuesto utilizando clasificadores globales. Los tres valores de F más altos se resaltan en negrita (1º), azul (2º) y rojo (3º). En la última columna se muestra, a modo de comparación, el mejor resultado obtenido con detección de anomalías. En la Tabla A.1 se recogen los resultados detallados para cada *tracker*.

			Método (Clasificadores)										Detección de anomalías (<i>covar. matrix differ.</i>)
			<i>Boosting</i> (Árboles)							<i>Bagging</i> (Árboles)	Subesp. aleat. (KNN)	Subesp. aleat. (Discr.)	
			<i>Ada BoostM1</i>	<i>Logit Boost</i>	<i>Gentle Boost</i>	<i>Robust Boost</i>	<i>LP Boost</i>	<i>Total Boost</i>	<i>RUS Boost</i>				
Media	Detección <i>frames</i> de cambio	P	0.395	0.523	0.551	0.538	0.353	0.208	0.083	0.560	0.278	0.490	0.462
		R	0.155	0.186	0.172	0.164	0.605	0.642	0.858	0.212	0.180	0.486	0.549
		F	0.204	0.248	0.228	0.231	0.241	0.254	0.148	0.294	0.170	0.458	0.489
	Decisión CT/WT	TPR	0.993	0.979	0.988	0.981	0.893	0.920	0.877	0.950	0.923	0.938	0.935
		FPR	0.889	0.756	0.869	0.807	0.361	0.292	0.142	0.677	0.481	0.481	0.359
		AUC	0.552	0.612	0.560	0.587	0.767	0.814	0.867	0.637	0.721	0.729	0.788

Tabla 5.10: Valor medio de los resultados obtenidos con el algoritmo propuesto utilizando métodos de combinación de clasificadores débiles. Los tres valores de F más altos se resaltan en negrita (1º), azul (2º) y rojo (3º). En la última columna se muestra, a modo de comparación, el mejor resultado obtenido con detección de anomalías. En la Tabla A.2 se recogen los resultados detallados para cada *tracker*.

5.7. Comparativa con el estado del arte

En esta sección se lleva a cabo una comparativa con dos de las técnicas del estado del arte analizadas en el Capítulo 2: clasificación CT/WT con *Naive Bayes* [11] y *backtracking*⁴ [12]. En dicha comparativa no se ha incluido ARTE porque únicamente es aplicable a *trackers* multi-hipótesis y no todos los *trackers* que se emplean en este trabajo lo son. Por tanto, los resultados que se obtendrían para este algoritmo serían parciales.

Para realizar la comparativa se han usado solamente los resultados obtenidos con *covariance matrix difference*, ya que es la única característica que ha funcionado ligeramente mejor. En el algoritmo de clasificación CT/WT se han considerado dos posibles casos en el momento de llevar a cabo la evaluación: emplear el set de características propuesto en [11]; o usar el set de características seleccionado en este trabajo (en el cual se incluyen las todas características de [11]

⁴En la comparativa ha de tenerse en cuenta que *backtracking* es una técnica de naturaleza distinta al algoritmo de clasificación CT/WT. El algoritmo propuesto ha sido desarrollado a partir de este último, por lo que la filosofía que comparten es más parecida.

y algunas más relacionadas con la apariencia del objeto). Para evaluar *backtracking* se ha hecho uso de cadenas de Markov en el caso de CPF. En el resto de *trackers* se ha aplicado la misma idea y se ha llevado a cabo un seguimiento en sentido inverso hasta el punto de referencia en donde se ha medido el solape espacial entre la imagen correcta del objeto y la imagen estimada por el *tracker backward*. En todos los casos se ha escogido como punto de referencia el *frame* inicial. Para reducir el coste computacional se ha realizado el análisis *backward* solamente cada 10 *frames*.

En la Tabla 5.11 se presentan los resultados de cada una de las técnicas y en la Figura 5.9 se muestran las curvas ROC. Se puede apreciar que el valor de FPR obtenido con el algoritmo de clasificación CT/WT es muy elevado en todos los *trackers*, lo que conlleva que el valor de AUC sea demasiado bajo. Esto ocurre en los dos casos planteados. El mal funcionamiento de este algoritmo se debe a que el clasificador no es capaz de distinguir entre las dos clases y predice, casi siempre, CT. Como ya se vio en la Sección 5.5.1, el hecho de modelar CT y WT por separado no es una opción correcta puesto que el parecido entre ambas clases es muy alto. El comportamiento de *backtracking*, en cambio, es más aceptable. Se observa que el valor de AUC obtenido con esta técnica es alto en todos los *trackers*. El algoritmo propuesto, por su parte, mejora en todos los casos el valor de FPR obtenido con el algoritmo de clasificación CT/WT y, como consecuencia, el valor de AUC. En general, sus resultados se aproximan bastante a los de *backtracking*.

Por otro lado se realiza también una comparación del coste computacional de las tres técnicas (ver Tabla 5.12). Para medir el coste del algoritmo propuesto y de la técnica de clasificación CT/WT se ha distinguido la fase de entrenamiento de la fase de test. Se comprueba que el coste computacional del algoritmo propuesto es con diferencia el más bajo, ya que necesita calcular una sola característica. La técnica de clasificación CT/WT emplea varias características, lo que supone un coste notablemente mayor. Los requerimientos computacionales de *backtracking* son asimismo muy elevados, sobre todo en SPT y LOT. Estos dos *trackers* son demasiado lentos, lo cual hace que el análisis *backward* se vuelva excesivamente costoso, a pesar de que se ha llevado a cabo sólo cada 10 *frames*. En el caso de CPF, en el que se han utilizado cadenas de Markov, se aprecia que la complejidad computacional es también bastante alta.

Como resumen de los resultados de cada una de las técnicas, se muestra en la Figura 5.10 una representación del coste computacional frente a AUC. Esta gráfica permite extraer dos conclusiones:

- El algoritmo propuesto consigue un valor de AUC bastante más alto que la técnica de clasificación CT/WT y además con un coste menor. Es decir, se logran mejorar claramente los resultados de la técnica que se ha utilizado como referencia.
- El algoritmo propuesto consigue un valor de AUC muy similar al obtenido con la técnica de *backtracking*. Sin embargo, el coste de *backtracking* es demasiado elevado.

			Algoritmo				%Δ mejor
			Clasificación CT/WT con <i>Naive Bayes</i>		<i>Backtracking</i> [12]	Detección de anomalías con <i>covar. matrix differ.</i> (propuesto)	
Tracker	MS	TPR	0.905	0.935	0.766	0.976	7.8
		FPR	0.807	0.821	0.108	0.270	150.0
		AUC	0.549	0.557	0.976	0.853	-12.6
	CPF	TPR	0.926	0.915	0.730	0.814	-12.1
		FPR	0.799	0.739	0.326	0.384	17.8
		AUC	0.564	0.588	0.735	0.715	-2.7
	LOT	TPR	0.976	0.964	0.903	0.974	-0.2
		FPR	0.955	0.923	0.080	0.511	538.8
		AUC	0.510	0.521	0.976	0.732	-25.0
	IVT	TPR	0.952	0.940	0.886	0.983	3.3
		FPR	0.555	0.448	0.092	0.304	230.4
		AUC	0.699	0.746	0.981	0.840	-14.4
	SPT	TPR	0.911	0.933	0.898	0.939	3.1
		FPR	0.593	0.592	0.129	0.450	248.8
		AUC	0.659	0.671	0.942	0.745	-20.9
	TLD	TPR	0.973	0.955	0.800	0.926	-4.8
		FPR	0.931	0.912	0.374	0.236	-36.9
		AUC	0.521	0.522	0.703	0.845	20.2
	Media	TPR	0.941	0.940	0.831	0.935	-0.6
		FPR	0.773	0.739	0.185	0.359	94.1
		AUC	0.584	0.601	0.886	0.788	-11.1

Tabla 5.11: Comparativa de los resultados conseguidos con el algoritmo propuesto y los obtenidos con dos técnicas del estado del arte: clasificación CT/WT con *Naive Bayes* y *backtracking*. En negrita se resaltan los mejores resultados. La última columna muestra, para cada uno de los resultados conseguidos con el algoritmo propuesto, su variación porcentual respecto al mejor de los resultados obtenidos usando las técnicas del estado del arte.

* En este caso se han utilizado todas las características seleccionadas en este trabajo (es decir, las propuestas en [11] junto con las que se han añadido relativas a la apariencia del objeto).

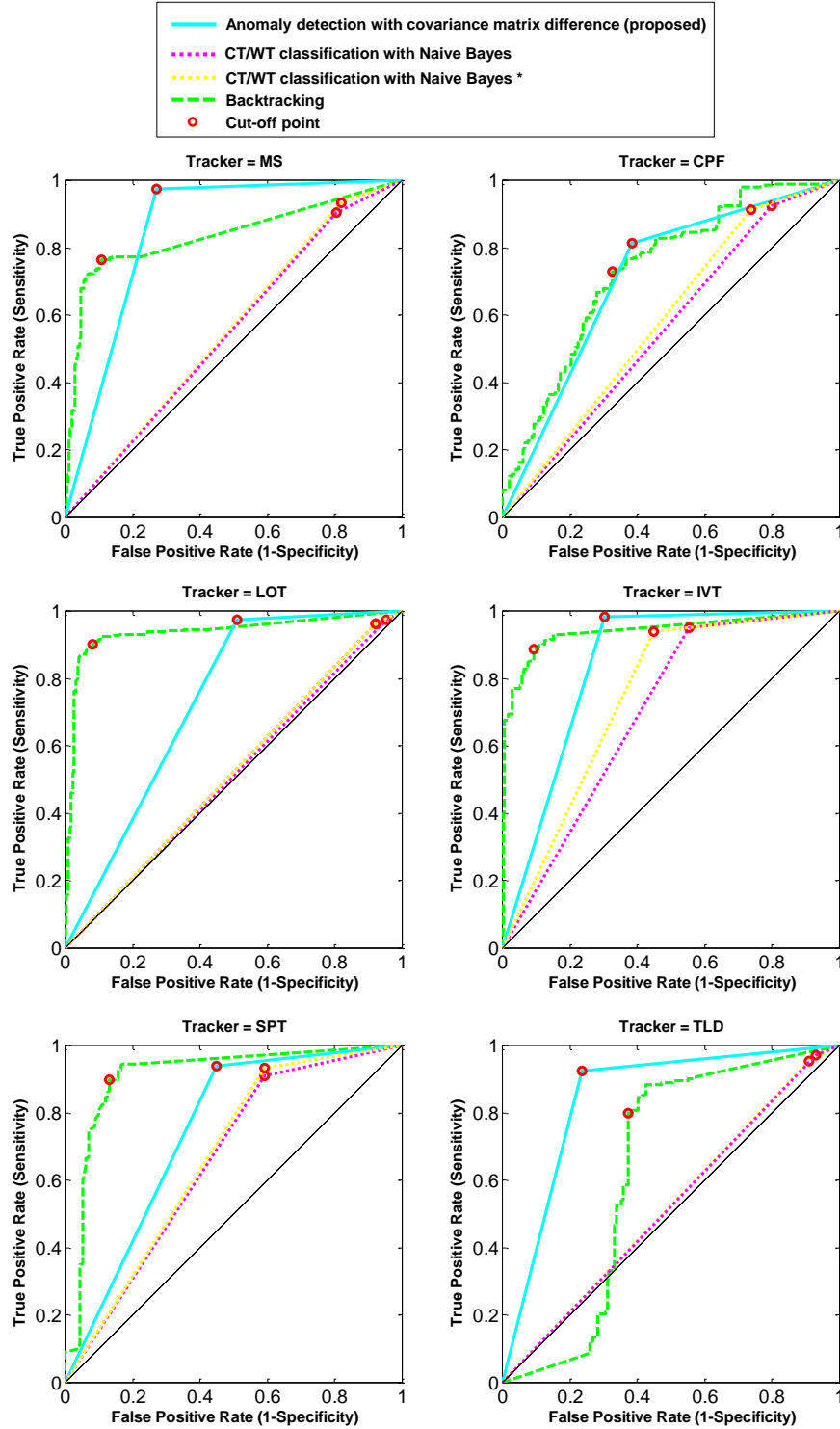


Figura 5.9: Curvas ROC para la segmentación temporal CT/WT utilizando el algoritmo propuesto y dos técnicas del estado del arte: clasificación CT/WT con *Naive Bayes* y *backtracking*. * En este caso se han utilizado todas las características seleccionadas en este trabajo (es decir, las propuestas en [11] junto con las que se han añadido relativas a la apariencia del objeto).

		Algoritmo							%Δ mejor (Test)
		Clasificación CT/WT con <i>Naive Bayes</i>				<i>Backtracking</i> [12]	Detección de anomalías con <i>covar. matrix differ.</i> (propuesto)		
		[11]		[11]*		Test	Entrenam.	Test	
Tracker	MS	4185	3513	4473	3807	3925	515	476	-87.3
	CPF	4515	4102	4885	4353	16513	613	640	-84.4
	LOT	4719	4331	4972	4527	37618	675	623	-85.6
	IVT	4108	3996	4298	4225	4510	533	576	-85.5
	SPT	4312	4055	4576	4360	93203	558	488	-88.0
	TLD	3955	3822	4266	4107	4318	506	450	-88.2
	Media	4299	3970	4578	4230	26681	567	542	-86.3

Tabla 5.12: Comparativa del coste computacional (ms/frame) del algoritmo propuesto y de dos técnicas del estado del arte: clasificación CT/WT con *Naive Bayes* y *backtracking*. En negrita se resalta el coste más bajo. La última columna muestra, para cada coste del algoritmo propuesto, su variación porcentual respecto al coste más bajo de las técnicas del estado del arte. Para esta comparación se han considerado únicamente las fases de test.

* En este caso se han utilizado todas las características seleccionadas en este trabajo (es decir, las propuestas en [11] junto con las que se han añadido relativas a la apariencia del objeto).

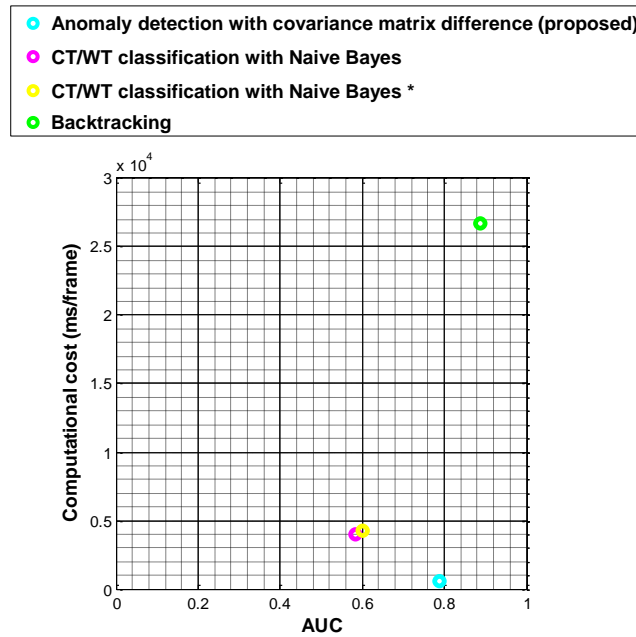


Figura 5.10: Coste computacional (ms/frame) frente a AUC para el algoritmo propuesto y para dos técnicas del estado del arte: clasificación CT/WT con *Naive Bayes* y *backtracking*.

* En este caso se han utilizado todas las características seleccionadas en este trabajo (es decir, las propuestas en [11] junto con las que se han añadido relativas a la apariencia del objeto).

Capítulo 6

Conclusiones y trabajo futuro

En este capítulo se ofrece un resumen del trabajo realizado (Sección 6.1), se exponen las conclusiones finales (Sección 6.2) y se plantean futuras líneas de investigación (Sección 6.3).

6.1. Resumen

En este proyecto se ha presentado una técnica para la estimación de fiabilidad de los algoritmos de seguimiento de objetos en vídeo (*trackers*).

Para comenzar se realizó una revisión del estado del arte referente a la estimación de fiabilidad. Se comprobó que se trata de un área que requiere ser mejorada ya que los métodos existentes tienen bastantes limitaciones. Por lo general, la mayor parte son dependientes de la aplicación; algunos son computacionalmente irrealizables si las secuencias son largas; otros son poco robustos ante errores del *tracker*.

Además se examinaron más detalladamente tres propuestas de estimación de fiabilidad recientes: clasificación CT/WT con *Naive Bayes*, *backtracking* utilizando cadenas de Markov y ARTE. La primera de estas técnicas plantea combinar en un clasificador *Naive Bayes* un conjunto de características relacionadas con la forma, el movimiento y la apariencia del objeto seguido para determinar en cada momento si el seguimiento es correcto (CT) o incorrecto (WT). Sin embargo se observó que dichas características adquieren valores muy parecidos en ambos casos y el clasificador no es capaz de distinguir entre CT y WT. En este punto se llegó a la conclusión de que lo más apropiado era emplearlas para detectar los *frames* de cambio en los que el *tracker* pierde o recupera al objeto, ya que en estos instantes tienden a tomar unos valores más anómalos. Esto motivó el desarrollo de un nuevo algoritmo que hiciera uso de estas características, a las que se añadieron algunas más relativas a la apariencia. Se decidió apostar entonces por una estrategia de detección de anomalías para discriminar esos valores atípicos de las características. Se plantearon tres versiones de dicha estrategia: *un modelo de CT por cada característica*, *un modelo de CT para todas las características* y *combinación de las detecciones de cada característica*. Asimismo

se consideró la utilización de una máquina de estados para realizar la segmentación temporal CT/WT una vez detectados los posibles *frames* de cambio.

La evaluación del algoritmo propuesto se llevó a cabo sobre seis *trackers* empleando el *data-set* público SOVTds que incluye los problemas de seguimiento más frecuentes que aparecen en escenarios reales. Por último, los resultados conseguidos con el algoritmo se compararon con los obtenidos con la técnica de clasificación CT/WT y con *backtracking*.

6.2. Conclusiones

Las conclusiones que se han extraído tras la realización del proyecto son las siguientes:

- El uso de las características en [11] para realizar la segmentación temporal CT/WT no es el adecuado. Se ha comprobado que todas ellas toman valores muy similares cuando el seguimiento es correcto y cuando el seguimiento es incorrecto, por lo que el clasificador *Naive Bayes* no puede diferenciar CT y WT.
- De todas las características que se han planteado en este trabajo para detectar los *frames* de cambio la única que ha tenido un comportamiento más aceptable ha sido *covariance matrix difference*, que en todos los *trackers* alcanzó el mayor valor de F. Esta característica está intrínsecamente relacionada con la estructura de los niveles de intensidad del objeto que a corto plazo debe ser estable.
- Los resultados del algoritmo propuesto con *covariance matrix difference* han conseguido mejorar a los obtenidos con la técnica de clasificación CT/WT (la cual ha servido como punto de partida de este trabajo) y se han aproximado bastante a los logrados por la técnica de *backtracking*. Además el coste computacional del algoritmo propuesto ha sido significativamente más bajo en todos los casos.

6.3. Trabajo futuro

La estimación de fiabilidad del seguimiento de objetos en vídeo continúa siendo a día de hoy un desafío abierto a futuras líneas de investigación. En lo que se refiere a este proyecto se sugieren las siguientes:

- **Mejorar la precisión de las características empleadas.** Sería deseable desarrollar nuevas características que sean más robustas frente a los distintos problemas de seguimiento observados.
- **Incluir ventanas temporales de análisis para localizar los *frames* de cambio.** Esto permitiría identificar pérdidas y recuperaciones lentas que, normalmente, son difíciles de detectar entre *frames* consecutivos.

- **Replantear el algoritmo y combinarlo con la técnica de *backtracking*.** Por ejemplo, se podría llevar a cabo un seguimiento hacia atrás en los *frames* de cambio detectados para verificar si efectivamente el *tracker* ha perdido/recuperado al objeto.

Bibliografía

- [1] E. Maggio and A. Cavallaro. *Video tracking: theory and practice*. Wiley, 2011. 1
- [2] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song. Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74(18):3823–3831, 2011. 1
- [3] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99:1, 2013. 1
- [4] A. Baumann, M. Boltz, J. Ebling, M. Koenig, H. S. Loos, M. Merkel, W. Niem, J. K. Warzelhan, and J. Yu. A review and comparison of measures for automatic video surveillance systems. *EURASIP Journal on Image and Video Processing*, 2008(824726):1–30, 2008. 1
- [5] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang. Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):319–336, 2009. 1
- [6] J. C. Nascimento and J. S. Marques. Performance evaluation of object detection algorithms for video surveillance. *IEEE Transactions on Multimedia*, 8(4):761–774, 2006. 1
- [7] T. Schlögl, C. Beleznai, M. Winter, and H. Bischof. Performance evaluation metrics for motion detection and tracking. In *International Conference on Pattern Recognition*, volume 4, pages 519–522, 2004. 1
- [8] J. C. San Miguel, A. Cavallaro, and J. M. Martínez. Adaptive online performance evaluation of video trackers. *IEEE Transactions on Image Processing*, 21(5):2812–2823, 2012. <http://www-vpu.eps.uam.es/publications/TrackQuality/>. 3, 5, 7, 10, 11, 12, 13, 17
- [9] J. C. San Miguel, A. Cavallaro, and J. M. Martínez. Standalone evaluation of deterministic video tracking. In *19th IEEE International Conference on Image Processing*, pages 1353–1356. IEEE, 2012. 3
- [10] J. C. San Miguel, A. Cavallaro, and J. M. Martínez. Evaluation of on-line quality estimators for object tracking. In *17th IEEE International Conference on Image Processing*, pages 825–828. IEEE, 2010. 3
- [11] C. Spampinato, S. Palazzo, and D. Giordano. Evaluation of tracking algorithm performance without ground-truth data. In *19th IEEE International Conference on Image Processing*, pages 1345–1348, 2012. 3, 7, 13, 23, 24, 27, 36, 44, 56, 58, 59, 60, 62

- [12] H. Wu, A. C. Sankaranarayanan, and R. Chellappa. Online empirical evaluation of tracking algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1443–1458, 2010. 3, 6, 8, 9, 10, 56, 58, 60
- [13] A Single Object Video Tracking dataset. <http://www-vpu.eps.uam.es/DS/SOVTds/content.htm>. 3, 38
- [14] D. P. Chau, F. Bremond, and M. Thonnat. Online evaluation of tracking algorithm performance. In *3rd International Conference on Crime Detection and Prevention*, pages 1–6. IET, 2009. 5, 6
- [15] H. Wu and Q. Zheng. Self-evaluation for video tracking systems. Technical report, DTIC Document, 2004. 5, 6
- [16] A. Doulamis. Dynamic tracking re-adjustment: a method for automatic tracking recovery in complex visual environments. *Multimedia Tools and Applications*, 50(1):49–73, 2010. 5
- [17] J. Black, T. Ellis, and P. Rosin. A novel method for video tracking performance evaluation. In *International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 125–132, 2003. 5, 6
- [18] E. Polat, M. Yeasin, and R. Sharma. Tracking body parts of multiple people: a new approach. In *Proceedings of the IEEE Workshop on Multi-Object Tracking*, pages 35–42. IEEE, 2001. 5
- [19] Z. Kalal, K. Mikolajczyk, and J. Matas. Forward-backward error: Automatic detection of tracking failures. In *20th International Conference on Pattern Recognition*, pages 2756–2759. IEEE, 2010. 6
- [20] R. Liu, S. Z. Li, X. Yuan, and R. He. Online determination of track loss using template inverse matching. In *The Eighth International Workshop on Visual Surveillance*, 2008. 6
- [21] C. Piciarelli, G. L. Foresti, and L. Snidaro. Trajectory clustering and its applications for video surveillance. In *IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 40–45. IEEE, 2005. 6
- [22] D. Hall. Automatic parameter regulation of perceptual systems. *Image and Vision Computing*, 24(8):870–881, 2006. 6
- [23] R. T. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1631–1643, 2005. 6
- [24] Z. Han, Q. Ye, and J. Jiao. Online feature evaluation for object tracking using kalman filter. In *19th International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008. 6
- [25] Ç. E. Erdem, B. Sankur, and A. M. Tekalp. Performance measures for video object segmentation and tracking. *IEEE Transactions on Image Processing*, 13(7):937–951, 2004. 6
- [26] Ç. E. Erdem, A. M. Tekalp, and B. Sankur. Video object tracking with feedback of performance measures. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(4):310–324, 2003. 6
- [27] P. L. Correia and F. Pereira. Stand-alone objective segmentation quality evaluation. *EURASIP Journal on Applied Signal Processing*, 2002(1):389–400, 2002. 6

- [28] C. Motamed. Motion detection and tracking using belief indicators for an automatic visual-surveillance system. *Image and Vision Computing*, 24(11):1192–1201, 2006. 6
- [29] K. Nickels and S. Hutchinson. Estimating uncertainty in ssd-based feature tracking. *Image and vision computing*, 20(1):47–58, 2002. 6
- [30] N. Vaswani. Additive change detection in nonlinear systems with unknown change parameters. *IEEE Transactions on Signal Processing*, 55(3):859–872, 2007. 6
- [31] V. Badrinarayanan, P. Perez, F. Le Clerc, and L. Oisel. On uncertainties, random features and object tracking. In *IEEE International Conference on Image Processing*, volume 5, pages 61–64. IEEE, 2007. 6
- [32] L. Lu, X. Dai, and G. Hager. A particle filter without dynamics for robust 3D face tracking. In *Conference on Computer Vision and Pattern Recognition Workshop*, page 70. IEEE, 2004. 6
- [33] E. Maggio, F. Smerladi, and A. Cavallaro. Adaptive multifeature tracking in a particle filtering framework. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(10):1348–1359, 2007. 6
- [34] A. D. Bagdanov, A. del Bimbo, F. Dini, and W. Nunziati. Adaptive uncertainty estimation for particle filter-based trackers. In *14th International Conference on Image Analysis and Processing*, pages 331–336. IEEE, 2007. 6
- [35] F. Van der Heijden. Consistency checks for particle filters. *IEEE transactions on pattern analysis and machine intelligence*, 28(1):140–145, 2006. 6
- [36] R. M. Powers and L. Y. Pao. Power and robustness of a track-loss detector based on Kolmogórov-Smirnov tests. In *American Control Conference*, pages 8–pp. IEEE, 2006. 6
- [37] P. Pan, F. Porikli, and D. Schonfeld. A new method for tracking performance evaluation based on a reflective model and perturbation analysis. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3529–3532. IEEE, 2009. 6
- [38] I. Rish. An empirical study of the naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001. 7, 73
- [39] A. Doucet. *Sequential Monte Carlo methods*. Wiley Online Library, 2001. 8
- [40] P. C. Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55, 1936. 9
- [41] K. Nummiaro, E. Koller-Meier, and L. Van Gool. An adaptive color-based particle filter. *Image and vision computing*, 21(1):99–110, 2003. 10, 37
- [42] D. Schreiber. Robust template tracking with drift correction. *Pattern recognition letters*, 28(12):1483–1491, 2007. 17
- [43] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003. 18, 26, 37

- [44] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15, 2009. 19
- [45] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, 2012. <http://personal.ee.surrey.ac.uk/Personal/Z.Kalal/tld.html>. 24, 37
- [46] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan. Locally orderless tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1940–1947. IEEE, 2012. <http://www.eng.tau.ac.il/~oron/LOT/LOT.html>. 25, 37
- [47] R. Brunelli. *Template matching techniques in computer vision: theory and practice*. John Wiley & Sons, 2009. 27, 36
- [48] D. Gabor. Theory of communication. *Journal of Institute for Electrical Engineering, part III*, 93(26):429–457, 1946. 27
- [49] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing focuses on signal-processing aspects of image processing, imaging systems, and image scanning, display, and printing*, 13(4):600–612, 2004. <https://ece.uwaterloo.ca/~z70wang/research/ssim/>. 29, 31
- [50] A. Loza, L. Mihaylova, N. Canagarajah, and D. Bull. Structural similarity-based object tracking in video sequences. In *9th International Conference on Information Fusion*, pages 1–6. IEEE, 2006. 30, 36
- [51] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893. IEEE, 2005. 31, 36
- [52] B. Leibe and K. Grauman. Visual Object Recognition, 2008. http://www.cs.utexas.edu/~grauman/slides/aaai2008/aaai2008-tutorial-part2_sliding_windows.pdf. 32
- [53] A. Bhattacharyya. On a measure of divergence between two multinomial populations. *Bulletin of the Calcutta Mathematical Society*, 35:99–108, 1943. 32, 42
- [54] M. Oussalah and M. Shabash. Object tracking using level set and MPEG 7 color features. In *3rd International Conference on Image Processing Theory, Tools and Applications*, pages 105–110. IEEE, 2012. 32, 36
- [55] L. Cieplinski. MPEG-7 color descriptors and their applications. In *Computer analysis of images and patterns*, pages 11–20. Springer, 2001. 32
- [56] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based means on riemannian manifolds. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2006. 34, 35, 36
- [57] W. Förstner and B. Moonen. A metric for covariance matrices. In *Geodesy-The Challenge of the 3rd Millennium*, pages 299–309. Springer, 2003. 35

- [58] D. A. Ross, J. Lim, R. Lin, and M. Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008. <http://www.cs.toronto.edu/~dross/ivt/>. 37
- [59] S. Wang, H. Lu, F. Yang, and M. Yang. Superpixel tracking. In *IEEE International Conference on Computer Vision*, pages 1323–1330. IEEE, 2011. 37
- [60] F. J. Massey Jr. The Kolmogórov-Smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951. 44
- [61] K. Pearson. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58(347-352):240–242, 1895. 45
- [62] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967. 73
- [63] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 1998. 73
- [64] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and regression trees*. Wadsworth and Brooks, 1984. 73
- [65] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *Annals of statistics*, 28:337–374, 2000. 73
- [66] Y. Freund. A more robust boosting algorithm. *arXiv preprint arXiv:0905.2138*, 2009. 73
- [67] M. K Warmuth, J. Liao, and G. Rätsch. Totally corrective boosting algorithms that maximize the margin. In *Proceedings of the 23rd international conference on Machine learning*, pages 1001–1008. ACM, 2006. 73
- [68] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano. Rusboost: Improving classification performance when training data is skewed. In *19th International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008. 73
- [69] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996. 73
- [70] T. K. Ho. The random subspace method for constructing decision forests. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(8):832–844, 1998. 73
- [71] MIT Traffic dataset. <http://www.ee.cuhk.edu.hk/~xgwang/MITtraffic.html>. 80
- [72] Institut für Algorithmen und Kognitive Systeme. http://i21www.ira.uka.de/image_sequences/. 80
- [73] TREC Video Retrieval Evaluation. trecvid.nist.gov/trecvid.data.html. 80
- [74] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 232–237. IEEE, 1998. <http://www.ces.clemson.edu/~stb/research/headtracker/seq/>. 80
- [75] R. Vezzani and R. Cucchiara. Video surveillance online repository (visor): an integrated framework. *Multimedia Tools and Applications*, 50(2):359–380, 2010. 80

- [76] Imagery Library for Intelligent Detection Systems. <https://www.gov.uk/imagery-library-for-intelligent-detection-systems>. 80
- [77] 11th IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance, 2009. <http://www.cvg.rdg.ac.uk/PETS2009/>. 80
- [78] 1st IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance, 2000. <ftp://ftp.pets.rdg.ac.uk/pub/PETS2000/>. 80
- [79] Context Aware Vision using Image-based Active Recognition. <http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/>. 80

Apéndice A

Otras estrategias para la detección de los *frames* de cambio

A.1. Clasificación supervisada

La clasificación supervisada se basa en la disponibilidad de un conjunto de datos de entrenamiento de los cuales se conoce a priori la clase a la que pertenecen y que sirven para generar un modelo que permitirá clasificar nuevos datos. En este trabajo la idea es entrenar diversas técnicas de clasificación supervisada con vectores de características para que sean capaces de discriminar los *frames* de cambio del resto de *frames*. Se plantean dos tipos de estrategias:

- **Clasificadores globales.** Se consideran las técnicas *Naive Bayes* [38], *Nearest Neighbors* [62], *Support Vector Machine* (SVM) [63], *Classification Trees* [64] y *Regression Trees* [64]. Cada una utiliza un procedimiento distinto para llevar a cabo la clasificación.
- **Combinación de clasificadores débiles.** El objetivo es utilizar métodos ensambladores que consisten en combinar un conjunto de clasificadores débiles para crear un clasificador más robusto que consiga una mayor precisión. La principal ventaja de los clasificadores débiles es la rapidez de su entrenamiento. Concretamente se van a explorar los métodos de *Boosting* (*AdaBoostM1* [65], *LogitBoost* [65], *GentleBoost* [65], *RobustBoost* [66], *LPBoost* [67], *TotalBoost* [67] y *RUSBoost* [68]) y *Bagging* [69], que hacen uso de árboles de decisión y el método de Subespacio aleatorio [70] que puede utilizar las técnicas *K-Nearest Neighbors* (KNN) o análisis discriminante.

Para cada uno de los clasificadores se definen, por tanto, dos clases: por un lado, los *frames* de cambio (C) y, por otro, los *frames* en los que no hay cambio (NC). La clasificación se lleva a cabo en dos fases (ver Figura A.1):

- **Fase de entrenamiento.** La primera fase consiste en entrenar al clasificador con vectores de características a los que se les asigna una etiqueta de clase (C/NC). Para ello se emplea un grupo de secuencias de entrenamiento y se hace uso de las anotaciones de GT para determinar la etiqueta de cada vector. Se crea así un modelo que se empleará para clasificar nuevos vectores de características.
- **Fase de test.** En las secuencias de test el clasificador procesa el vector de características $\mathbf{f}_t = (f_t^{(1)}, \dots, f_t^{(n)})$ asociado a cada *frame* t y decide a cuál de las dos clases pertenece en función de la probabilidad p obtenida a partir del modelo generado. Es decir, detecta si se trata de un *frame* de cambio o no. La señal de cambio será entonces:

$$change_t = \begin{cases} 1 & \text{si } p_C(\mathbf{f}_t) > p_{NC}(\mathbf{f}_t) \\ 0 & \text{resto} \end{cases} \quad (\text{A.1})$$

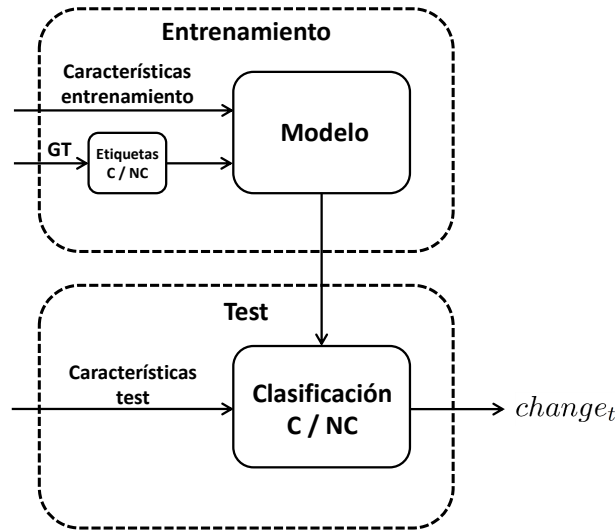


Figura A.1: Diagrama de bloques genérico de las técnicas de clasificación supervisada.

A.1.1. Evaluación del algoritmo propuesto

En este apartado se lleva a cabo la evaluación del algoritmo propuesto empleando para la detección de los *frames* de cambio las distintas técnicas de clasificación supervisada. Las secuencias utilizadas en cada *tracker* han sido las mismas que en el caso de detección de anomalías: 101 para la fase de entrenamiento y 25 para la fase de test. Se han usado, en cada *tracker*, las dos características que individualmente consiguieron el mayor valor de F en la técnica de detección de anomalías. En la máquina de estados se ha utilizado también *covariance matrix difference*

para realizar la comparación con la imagen del objeto en el *frame* inicial

En la Tabla A.1 se presentan los resultados que se han obtenido con cada uno de los clasificadores globales. Se puede ver que la mejor puntuación la consigue, en casi todos los *trackers*, el clasificador *Naive Bayes*, alcanzando los valores más altos en MS y TLD.

En la Tabla A.2 se exponen los resultados obtenidos con los diferentes métodos de combinación de clasificadores débiles. Se comprueba que, en general, el método que mejores resultados ofrece es el de Subespacio aleatorio con técnicas de análisis discriminante. También se observa que las puntuaciones más altas se alcanzan en MS, TLD y CPF.

			Clasificador				
			<i>Naive Bayes</i>	<i>Nearest Neighbors</i>	<i>Support Vector Machine</i>	<i>Classification Trees</i>	<i>Regression Trees</i>
<i>Tracker</i>	MS	Características usadas	<i>covariance matrix difference</i> y <i>HOG difference</i>				
		Detección <i>frames</i> de cambio	P	0.640	0.947	1.000	0.743
			R	0.744	0.419	0.349	0.605
			F	0.688	0.581	0.517	0.648
		Decisión CT/WT	TPR	0.936	0.984	0.980	0.964
			FPR	0.472	0.649	0.727	0.594
			AUC	0.732	0.668	0.627	0.684
	CPF	Características usadas	<i>covariance matrix difference</i> y <i>CLD difference</i>				
		Detección <i>frames</i> de cambio	P	0.900	0.320	0.305	0.920
			R	0.429	0.254	0.841	0.111
			F	0.581	0.283	0.447	0.523
		Decisión CT/WT	TPR	0.782	0.766	0.777	0.821
			FPR	0.436	0.400	0.182	0.452
			AUC	0.673	0.683	0.798	0.671
	LOT	Características usadas	<i>covariance matrix difference</i> e <i>histogram difference grayscale</i>				
		Detección <i>frames</i> de cambio	P	1.000	0.148	-	0.222
			R	0.032	0.129	0.000	0.000
			F	0.063	0.138	-	0.100
		Decisión CT/WT	TPR	1.000	0.997	1.000	0.998
			FPR	1.000	0.736	1.000	0.950
			AUC	0.500	0.631	0.500	0.523
	IVT	Características usadas	<i>covariance matrix difference</i> y <i>DSSIM</i>				
		Detección <i>frames</i> de cambio	P	0.429	0.136	0.175	0.222
			R	0.120	0.120	0.440	0.080
			F	0.188	0.127	0.250	0.118
		Decisión CT/WT	TPR	1.000	0.922	0.983	0.990
			FPR	0.677	0.620	0.304	0.673
			AUC	0.662	0.651	0.840	0.659
	SPT	Características usadas	<i>covariance matrix difference</i> y <i>HOG difference</i>				
		Detección <i>frames</i> de cambio	P	0.667	0.278	0.062	0.500
			R	0.114	0.143	0.371	0.086
			F	0.195	0.189	0.107	0.146
		Decisión CT/WT	TPR	0.993	0.952	0.909	0.956
			FPR	0.782	0.472	0.274	0.629
			AUC	0.606	0.740	0.818	0.664
	TLD	Características usadas	<i>covariance matrix difference</i> y <i>CLD difference</i>				
		Detección <i>frames</i> de cambio	P	0.615	0.739	0.938	0.600
			R	0.727	0.386	0.341	0.409
			F	0.667	0.508	0.500	0.487
		Decisión CT/WT	TPR	0.918	0.986	0.988	0.961
			FPR	0.224	0.346	0.703	0.339
			AUC	0.847	0.820	0.643	0.811
	Media	Detección <i>frames</i> de cambio	P	0.709	0.428	0.413	0.535
			R	0.361	0.242	0.390	0.268
			F	0.397	0.304	0.304	0.340
		Decisión CT/WT	TPR	0.938	0.935	0.940	0.948
			FPR	0.599	0.537	0.532	0.599
			AUC	0.670	0.699	0.704	0.675

Tabla A.1: Resultados obtenidos con el algoritmo propuesto utilizando clasificadores globales. Se han usado, en cada *tracker*, las dos características que individualmente consiguieron las mejores puntuaciones en la técnica de detección de anomalías. Los tres valores de F más altos se resaltan en negrita (1º), azul (2º) y rojo (3º).

			Método (Clasificadores)										
			Boosting (Árboles)							Bagging (Árboles)	Subesp. aleat. (KNN)	Subesp. aleat. (Discr.)	
			Ada Boost M1	Logit Boost	Gentle Boost	Robust Boost	LP Boost	Total Boost	RUS Boost				
Tracker	MS	Características usadas	covariance matrix difference y HOG difference										
		Detección frames de cambio	P	0.867	0.929	0.963	0.958	0.154	0.762	0.098	0.750	0.154	0.600
			R	0.605	0.605	0.605	0.535	0.884	0.744	0.930	0.488	0.279	0.767
			F	0.712	0.732	0.743	0.687	0.262	0.753	0.177	0.592	0.198	0.674
		Decisión CT/WT	TPR	0.964	0.969	0.975	0.975	0.941	0.976	0.929	0.975	0.989	0.948
			FPR	0.606	0.606	0.606	0.607	0.210	0.347	0.171	0.592	0.533	0.266
			AUC	0.679	0.682	0.685	0.684	0.866	0.815	0.879	0.692	0.728	0.841
	CPF	Características usadas	covariance matrix difference y CLD difference										
		Detección frames de cambio	P	-	0.857	1.000	0.833	0.675	0.148	0.066	0.790	0.345	0.683
			R	0.000	0.095	0.032	0.079	0.428	0.857	0.984	0.238	0.318	0.651
			F	-	0.171	0.062	0.145	0.524	0.252	0.124	0.366	0.331	0.667
		Decisión CT/WT	TPR	1.000	0.915	0.964	0.924	0.804	0.838	0.740	0.749	0.745	0.770
			FPR	1.000	0.576	0.999	0.576	0.384	0.200	0.088	0.472	0.306	0.519
			AUC	0.500	0.670	0.483	0.674	0.710	0.819	0.826	0.639	0.720	0.626
	LOT	Características usadas	covariance matrix difference e histogram difference grayscale										
		Detección frames de cambio	P	-	-	-	-	0.009	0.018	0.049	0.500	0.053	0.400
			R	0.000	0.000	0.000	0.000	0.871	0.774	0.774	0.065	0.129	0.065
			F	-	-	-	-	0.018	0.036	0.091	0.114	0.075	0.111
		Decisión CT/WT	TPR	1.000	1.000	1.000	1.000	0.902	0.929	0.889	0.996	0.938	0.985
			FPR	1.000	1.000	1.000	1.000	0.227	0.311	0.174	0.912	0.554	0.794
			AUC	0.500	0.500	0.500	0.500	0.838	0.809	0.858	0.542	0.692	0.596
	IVT	Características usadas	covariance matrix difference y DSSIM										
		Detección frames de cambio	P	-	-	-	-	0.008	0.051	0.049	0.250	0.047	0.250
			R	0.000	0.000	0.000	0.000	0.960	0.200	0.840	0.040	0.080	0.400
			F	-	-	-	-	0.016	0.081	0.093	0.069	0.059	0.308
		Decisión CT/WT	TPR	1.000	1.000	1.000	1.000	0.753	0.969	0.922	1.000	0.968	0.975
			FPR	1.000	1.000	1.000	1.000	0.060	0.371	0.038	0.896	0.574	0.303
			AUC	0.500	0.500	0.500	0.500	0.847	0.799	0.942	0.552	0.697	0.836
	SPT	Características usadas	covariance matrix difference y HOG difference										
		Detección frames de cambio	P	0.500	0.500	0.500	0.500	0.500	0.016	0.047	0.222	0.066	0.409
			R	0.029	0.029	0.029	0.029	0.029	0.571	0.686	0.057	0.114	0.257
			F	0.054	0.054	0.054	0.054	0.054	0.030	0.088	0.091	0.083	0.316
		Decisión CT/WT	TPR	1.000	1.000	1.000	1.000	1.000	0.888	0.895	0.993	0.942	0.991
			FPR	0.956	0.956	0.956	0.956	0.956	0.249	0.187	0.826	0.378	0.782
			AUC	0.522	0.522	0.522	0.522	0.522	0.820	0.854	0.584	0.782	0.605
	TLD	Características usadas	covariance matrix difference y CLD difference										
		Detección frames de cambio	P	1.000	0.850	0.842	0.938	0.769	0.250	0.187	0.850	1.000	0.597
			R	0.296	0.386	0.364	0.341	0.455	0.705	0.932	0.386	0.159	0.773
			F	0.456	0.531	0.508	0.500	0.571	0.369	0.312	0.531	0.275	0.673
		Decisión CT/WT	TPR	0.995	0.988	0.988	0.988	0.960	0.917	0.885	0.987	0.958	0.959
			FPR	0.773	0.397	0.655	0.703	0.328	0.273	0.195	0.361	0.542	0.224
			AUC	0.611	0.796	0.667	0.643	0.816	0.822	0.845	0.813	0.708	0.868
	Media	Detección frames de cambio	P	0.395	0.523	0.551	0.538	0.353	0.208	0.083	0.560	0.278	0.490
			R	0.155	0.186	0.172	0.164	0.605	0.642	0.858	0.212	0.180	0.486
			F	0.204	0.248	0.228	0.231	0.241	0.254	0.148	0.294	0.170	0.458
		Decisión CT/WT	TPR	0.993	0.979	0.988	0.981	0.893	0.920	0.877	0.950	0.923	0.938
			FPR	0.889	0.756	0.869	0.807	0.361	0.292	0.142	0.677	0.481	0.481
			AUC	0.552	0.612	0.560	0.587	0.767	0.814	0.867	0.637	0.721	0.729

Tabla A.2: Resultados obtenidos con el algoritmo propuesto utilizando métodos de combinación de clasificadores débiles. Se han empleado, en cada *tracker*, las dos características que individualmente consiguieron las mejores puntuaciones en la técnica de detección de anomalías. Los tres valores de F más altos se resaltan en negrita (1º), azul (2º) y rojo (3º).

Apéndice B

SOVTds (*Single Object Video Tracking dataset*)

B.1. Factores de complejidad

En esta sección se definen los criterios empleados en el *dataset* SOVTds para estimar la complejidad de cada problema (ver Tabla B.1). Estos criterios incluyen factores objetivos (cambios de iluminación, oclusión y cambios de escala) y subjetivos (movimiento complejo, ruido y objetos similares). Algunos de estos factores se pueden generar artificialmente (ruido y cambios de iluminación), lo que permite crear secuencias sintéticas o modificar las reales con la complejidad requerida.

Problema	Criterio (factor)
Movimiento complejo	El objeto cambia su velocidad (píxels/ <i>frame</i>) de manera abrupta en <i>frames</i> consecutivos
Cambios de iluminación graduales	La intensidad media de un área cambia gradualmente con el tiempo hasta alcanzar una diferencia máxima de intensidad
Cambios de iluminación abruptos	La intensidad media de un área cambia abruptamente con respecto a sus alrededores (diferencia máxima de intensidad)
Ruido	Se incluye ruido natural (nieve) o ruido blanco gaussiano que se añade manualmente variando el valor de la desviación
Oclusión	En la escena aparecen objetos que ocultan en un porcentaje al objeto seguido
Cambios de escala	El objeto varía su tamaño con un cambio relativo máximo respecto a su tamaño original
Objetos similares	En las zonas próximas al objeto que se desea seguir aparece otro objeto del mismo color

Tabla B.1: Criterios y factores para estimar la complejidad de los problemas de seguimiento.

B.2. Situaciones modeladas

El *dataset* SOVTds está formado por 126 secuencias (~ 23000 *frames*) que se organizan en cuatro niveles (S1, S2, S3 y S4). Estos niveles varían desde situaciones completamente controladas (secuencias sintéticas) a situaciones no controladas (secuencias de entornos reales):

- **Secuencias sintéticas (S1).** El primer nivel está compuesto por un conjunto de secuencias sintéticas generadas artificialmente en unas condiciones controladas que permiten aislar cada problema. El objeto que se pretende seguir es una elipse morada que se mueve sobre un fondo negro que puede contener cuadrados también morados que actúan como objetos similares o de diferente color para simular oclusiones. Las secuencias creadas modelan todos los problemas de seguimiento con cinco grados de complejidad cada uno. En total hay 35 secuencias (~ 3500 *frames*).
- **Secuencias de laboratorio (S2).** El segundo nivel es una extensión natural del primero. En él se incluyen distintas secuencias grabadas bajo condiciones controladas en un laboratorio donde el objeto que se intenta seguir es una agenda azul. Las secuencias modelan todos los problemas de seguimiento con tres grados de complejidad cada uno. Algunos de los problemas (ruido y cambios de iluminación) se han añadido de manera artificial a secuencias grabadas previamente sin ningún problema. En total hay 21 secuencias (~ 6500 *frames*).
- **Secuencias reales simples (S3).** El tercer nivel está formado por clips de secuencias pertenecientes a otros *datasets* públicos ya existentes. Las secuencias han sido grabadas en entornos reales no controlados y contienen diferentes problemas de seguimiento aislados. Se agrupan en tres categorías dependiendo del tipo de objeto que se va a seguir: coches (de los *datasets* MIT Traffic [71] y Karlsruhe [72]), caras (de los *datasets* TRECVID2009 [73], CLEMSON [74] y VISOR [75]) y personas (de los *datasets* TRECVID2009 [73], i-Lids [76], PETS2009 [77], PETS2000 [78] y CAVIAR [79]). Para cada tipo de objeto y problema existen tres secuencias con distintos niveles de complejidad. En total hay 52 secuencias (~ 8500 *frames*).
- **Secuencias reales complejas (S4).** El cuarto nivel contiene, al igual que el anterior, clips extraídos de secuencias de *datasets* públicos ya existentes. Las secuencias han sido grabadas en entornos reales no controlados y en cada una hay una combinación de dos o más problemas de seguimiento. De la misma manera que en el tercer nivel, se distinguen tres categorías: coches (del *dataset* MIT Trac [71]), caras (del *dataset* CLEMSON [74]) y personas (del *dataset* PETS2009 [77]). En total hay 18 secuencias (~ 4500 *frames*).

Apéndice C

Presupuesto

1. Ejecución Material

- Compra de ordenador personal (Software incluido)2.000 €
- Alquiler de impresora láser durante 6 meses 260 €
- Material de oficina 150 €
- Total de ejecución material2.400 €

2. Gastos generales

- 16 % sobre Ejecución Material..... 352 €

3. Beneficio Industrial

- 6 % sobre Ejecución Material..... 132 €

4. Honorarios Proyecto

- 1800 horas a 15 €/ hora27.000 €

5. Material fungible

- Gastos de impresión.....280 €
- Encuadernación 200 €

6. Subtotal del presupuesto

- Subtotal Presupuesto.....32.774 €

7. I.V.A. aplicable

- 21 % Subtotal Presupuesto.....6.882,5 €

8. **Total presupuesto**

- Total Presupuesto 39.656,5 €

Madrid, julio de 2014

El Ingeniero Jefe de Proyecto

Fdo.: Álvaro Calvo Tapia

Ingeniero de Telecomunicación

Apéndice D

Pliego de condiciones

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un sistema para estimar la fiabilidad de algoritmos de seguimiento de objetos en vídeo. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.
2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.
3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.
4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.
6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.
7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.
8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.
9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.
10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.
11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partidaalzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.
13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.
14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.
15. La garantía definitiva será del 4 % del presupuesto y la provisional del 2 %.
16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.
17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.
18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.
19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.
20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.
21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.
23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrata" y anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

Condiciones particulares

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.
2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.
3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.
4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.
5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.
6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.
8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.
9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.
10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.
11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.
12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.